NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A239 523

DTIC
ELECTE
AUG 13 1991
S
B
D

THESIS

SCHEDULING AND ROUTING TACTICAL AERIAL
RECONNAISSANCE VEHICLES

by

Huey Douglas Moser, Jr.

September, 1990

Thesis Advisor: Richard E. Rosenthal

Approved for public release; distribution is unlimited.

91 8 12 047

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)		

6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL OR	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Including Security Classification)
SCHEDULING AND ROUTING TACTICAL AERIAL RECONNAISSANCE VEHICLES

12 PERSONAL AUTHOR(S)
MOSER, HUEY D., JR.

13 TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM TO	14. DATE OF REPORT (Year, Month, Day) 1990, September	15. Page Count 67

16. SUPPLEMENTAL NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Heuristic Programming, Traveling Salesman Probelm, Routing with time windows, Unmanned Aerial Vehicle.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
In this thesis we study the Marine Corps Tactical Aerial Reconnaissance Vehicle routing and scheduling problem The present method of routing and scheduling is presented, along with possible implications for routing and scheduling when future expansion of vehicle assets becomes available. A review of current literature is given, and comparisons are drawn between our problem and recent work. A model for the problem, which we call the Multi-Player Orienteering Problem with Time-Windows, is developed. We present both an optimization based solution and a heuristic solution for the problem. Computational results are shown for each, along with our reasons for selecting the heuristic solution as the best of the two solutions approaches attempted.

20 DISTRIBUTION/AVAILABILTIY OF ABSTRACT [X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC	1a. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Richard E. Rosenthal	22b. TELEPHONE (Include Area Code) (408)646-2792	22c. OFFICE SYMBOL OR/RI

Scheduling and Routing Tactical Aerial Reconnaissance Vehicles

by

Huey Douglas Moser, Jr.
Captain, United States Marine Corps
B.S., North Carolina State University, 1980

Submitted in partial fulfillment of the
requirements for the degree of

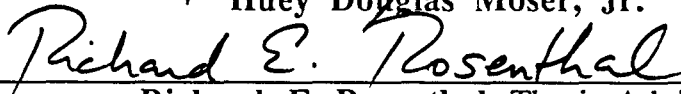**MASTER OF SCIENCE IN OPERATIONS RESEARCH**

from the

NAVAL POSTGRADUATE SCHOOL
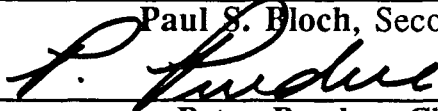**September 1990**

Author: _____
Huey Douglas Moser, Jr.

Approved by: _____
Richard. E. Rosenthal, Thesis Advisor

_____
Paul S. Bloch, Second Reader

_____
Peter Purdue, Chairman,
Department of Operations Research

ABSTRACT

In this thesis we study the Marine Corps Tactical Aerial Reconnaissance Vehicle routing and scheduling problem. The present method of routing and scheduling is presented, along with possible implications for routing and scheduling when future expansion of vehicle assets becomes available. A review of current literature is given, and comparisons are drawn between our problem and recent work. A model for the problem, which we call the *Multi-Player Orienteering Problem with Time-Windows*, is developed. We present both an optimization based solution and a heuristic solution for the problem. Computational results are shown for each, along with our reasons for selecting the heuristic solution as the best of the two solutions approaches attempted.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. THESIS CONTENT AND OVERVIEW

This thesis investigates the problem, faced by the United States Marine Corps and other U. S. military services, of efficiently routing and scheduling Tactical Aerial Reconnaissance (TACAIR RECCE) vehicles, and proposes a solution technique for the same. The present method of scheduling used by the Marine Corps is presented first, along with proposed research goals. Next, we give a brief survey of recent work in the routing and scheduling field. Following this survey, we explain the particular solution approaches attempted, and propose which of these we feel has the best opportunity for successful implementation. Computational experience with the solution chosen is presented, along with recommendations for implementation and possible future research opportunities.

B. PROBLEM BACKGROUND

1. Tactical Aerial Reconnaissance Systems

Tactical aerial reconnaissance (TACAIR RECCE) deals with the collection of data regarding the potential enemy's distribution and movement of forces, order of battle, and military actions through the use of tactical aerial vehicles and airborne sensor systems of intelligence gathering equipment. Closely aligned with TACAIR RECCE is the surveillance and observation missions, which focus on the placement of sensors in positions to observe and record data concerning either particular geographic areas or designated enemy possessions (equipment, personnel, etc.). In addition to use in intelligence activities (which are focused on the enemy),

1

reconnaissance missions can be undertaken so as to record certain blue force actions, such as collecting pre- or post-strike battle damage assessment (BDA).

During a TACAIR RECCE mission the data is collected by utilizing any combination of film-based sensors, modern electro-optical sensors, airborne radar and infrared sensors, and various types of electronic warfare intercept and eavesdropping devices. These sensors are carried aloft by various manned and unmanned vehicles. The manned vehicles used within the Marine Corps for TACAIR RECCE now and in the near future (five years hence) include the RF-4B Phantom II and the F/A- 18D Hornet aircraft. The unmanned aerial vehicles (UAV's) used and proposed for this mission are more numerous. Although at present there is only one type of UAV (the Pioneer system[1]) being utilized by Marine Corps units, the Department of Defense envisions the eventual incorporation of several other systems, in various stages of design and development for use by the Marines [Ref. 1]. These vehicles will have varying payload/range capabilities, with significant overlapping capabilities for redundancy and complementarity. Table 1 lists the various categories of vehicles along with potential sensor carrying capabilities. Figure 1. shows the range/endurance data for the various categories of TACAIR RECCE vehicles. Note that not all sensors can be carried by all vehicles.

---

[1]The word "system", when applied to tactical aerial reconnaissance, will refer throughout this paper to a vehicle and sensor combination.

TABLE 1. CATEGORY AND SENSOR TYPES OF TACAIR RECCE VEHICLES

Category	Sensor Types
Manned Aircraft RF-4B, F/A -18D	Film,(RF-4B only), Electro-Optical (EO),Radar, Infrared
Close Range UAV	EO, Electronic Intelligence (ELINT)
Short Range UAV	EO, Radar, ELINT, others
Medium Range UAV (Air-launched from manned aircraft)	EO, Infrared
Endurance UAV	ELINT,Radar, others

2. Present TACAIR RECCE Routing and Scheduling Method

The Marine Corps operates in accordance with a *"centralized command, decentralized control"* system for its aviation units. This means that command, and therefore tasking of units and aviation assets is decided at a central facility of headquarters, where the capabilities of each subunit can be most effectively coordinated with other units for maximum overall effectiveness. However, control of these subunits is executed at as low a level as possible consistent with sound tactical procedures. This system has been tested and adapted over the years to accommodate increases in electronic, radar, and communication capabilities.

As noted above, many of the missions assigned to TACAIR RECCE are in support of the intelligence gathering effort. The concept is for the overall commander to develop a series of questions about the enemies forces, disposition, movement, intentions, etc. These questions, along with questions posed by various staff officers who require information to complete their missions, are combined by the Intelligence staff into lists of questions called Essential Elements of Information
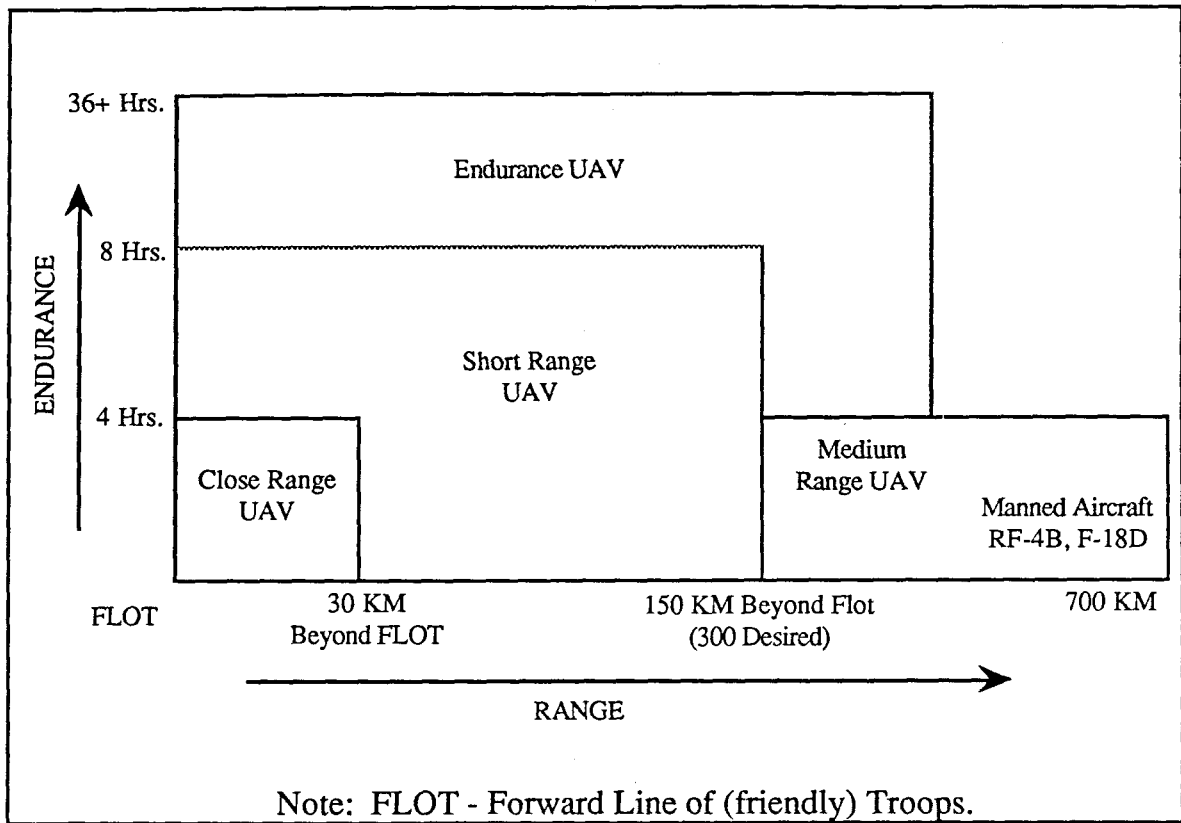
Figure 1. Range/Endurance For TACAIR RECCE Vehicles

(EEI's) and Other Elements of Information (OEI's). From these lists is derived a collection plan, which outlines the manner in which the needed data to answer the EEI's and OEI's will be obtained. One of the ways in which this data can be acquired is through the use of one or more of the TACAIR RECCE systems described earlier. Once this data is collected by these systems, it is combined with other data available and used to answer the EEI's and OEI's.

However, intelligence driven requirements are not the only possible sources of tasking for TACAIR RECCE assets. Other requirements might be generated by the Operations staff, as mentioned earlier. In developing the tasking for the TACAIR RECCE assets, the intelligence staff works with the operations staff

to decide what tasks might be most effectively satisfied by the TACAIR RECCE systems at the disposal of the command. Once it has been decided that a certain type of TACAIR RECCE vehicle might be able to satisfactorily perform a particular mission, a target designation is assigned to the area or place to be visited by the TACAIR RECCE vehicle and the new target is then assigned to one of the subunits[2] operating a suitable system. The target designation will include time requirements (no earlier than and no later than) on the targets, along with a service period required by the system to properly obtain the needed data. This type of information is usually referred to as a *time window* on the target. This process is continued until a determination is made by the operations staff that no new targets can be assigned to subunits. The two staffs will each have in mind a particular priority for each potential target, so that the requirements deemed most important will be given the most weight in target assignment, if it is not possible to satisfy all of the TACAIR RECCE vehicle taskings. This priority listing typically consists of groups of targets being lumped together as and designated as "highest", "high", "routine", etc. This prioritization is an informal mechanism used by the two staffs involved, but becomes extremely important in the cases where the command is limited in TACAIR RECCE systems available, which is nearly always the case. When assigning targets to the subunits, the operations staff of the higher headquarters, in conjunction with the intelligence staff, attempts to maximize potential benefit. In performing these assignments, target priority and geographic and temporal aspects are taken into account.

---

[2] These subunits might be squadrons (for manned aircraft), or companies or platoons (for UAV's).

Once all subunits' taskings have been set, an Air Tasking Order (ATO) is issued. The ATO contains target assignments and associated time requirements, along with coordinating instructions for the various subunits to allow them to function with higher and adjacent units more effectively. Once received by a subunit, the ATO is then translated into a daily flight schedule for the subunit by the schedules branch of the subunit's Operations Department. The schedules officer assigned to this task tries to reduce the cost of meeting the day's tasking when developing the schedule, while assuring that all assigned targets have been assigned to a specific TACAIR RECCE vehicle operated by the subunit. This process of producing the schedule is an exercise in routing, meeting the geographic constraint of the targets, and scheduling, meeting the time requirements of the targets. At times, the schedules officer is unable to develop a schedule that meets all target requirements. When this occurs, he then contacts the higher command operations staff, who originally assigned the targets to the subunits, for further guidance. This may mean a reduction in tasking for the subunit, or a change in targets already scheduled by the subunit so as to accommodate higher priority unscheduled targets. Once a completed schedule has been developed by the subunit, it is published and disseminated for subsequent use (generally, the next 12 to 24 hours). Figure 2 illustrates the process flow for target development, task assignment, and routing/scheduling activities.

```
┌──────────────────────────────────────────────────────────────────────┐
│  ┌─────────────────┐      ┌─────────────────┐          ╱‾‾‾‾‾‾‾╲       │
│  │ Intelligence Based│    │ Operations Based│          │ Start │       │
│  │   Requirements   │     │   Requirements  │          ╲_____╱        │
│  └─────────────────┘      └─────────────────┘                          │
│                                                                        │
│           ┌──────────────────────────┐                  ╱‾‾‾‾‾╲        │
│           │        Combined          │                  │ 6-12│        │
│           │      Requirements        │                  │ Hrs.│        │
│           └──────────────────────────┘                  ╲____╱         │
│                                                                        │
│           ┌──────────────────────────┐                                 │
│           │       Priorities,        │                                 │
│           │         System           │                                 │
│           │       Availability       │                                 │
│           └──────────────────────────┘                  ╱‾‾‾‾‾╲        │
│        Allocation of Targets to Subunits                │12-24│        │
│                                                         │ Hrs.│        │
│                                                         ╲____╱         │
│   ┌───────────┐    ┌───────────┐    ┌───────────┐                      │
│   │ Subunit 1 │    │ Subunit 2 │    │ Subunit 3 │                      │
│   └───────────┘    └───────────┘    └───────────┘                      │
│                                                                        │
│       Routing and Scheduling Process at Subunit                        │
│                                                                        │
│  ┌───────────┐    ┌───────────┐    ┌───────────┐        ╱‾‾‾‾‾╲        │
│  │ Completed │    │ Completed │    │ Completed │        │24-36│        │
│  │ Schedule  │    │ Schedule  │    │ Schedule  │        │ Hrs.│        │
│  │ Subunit 1 │    │ Subunit 1 │    │ Subunit 1 │        ╲____╱         │
│  └───────────┘    └───────────┘    └───────────┘                      │
└──────────────────────────────────────────────────────────────────────┘
```

Figure 2. Present Routing and Scheduling Development Process
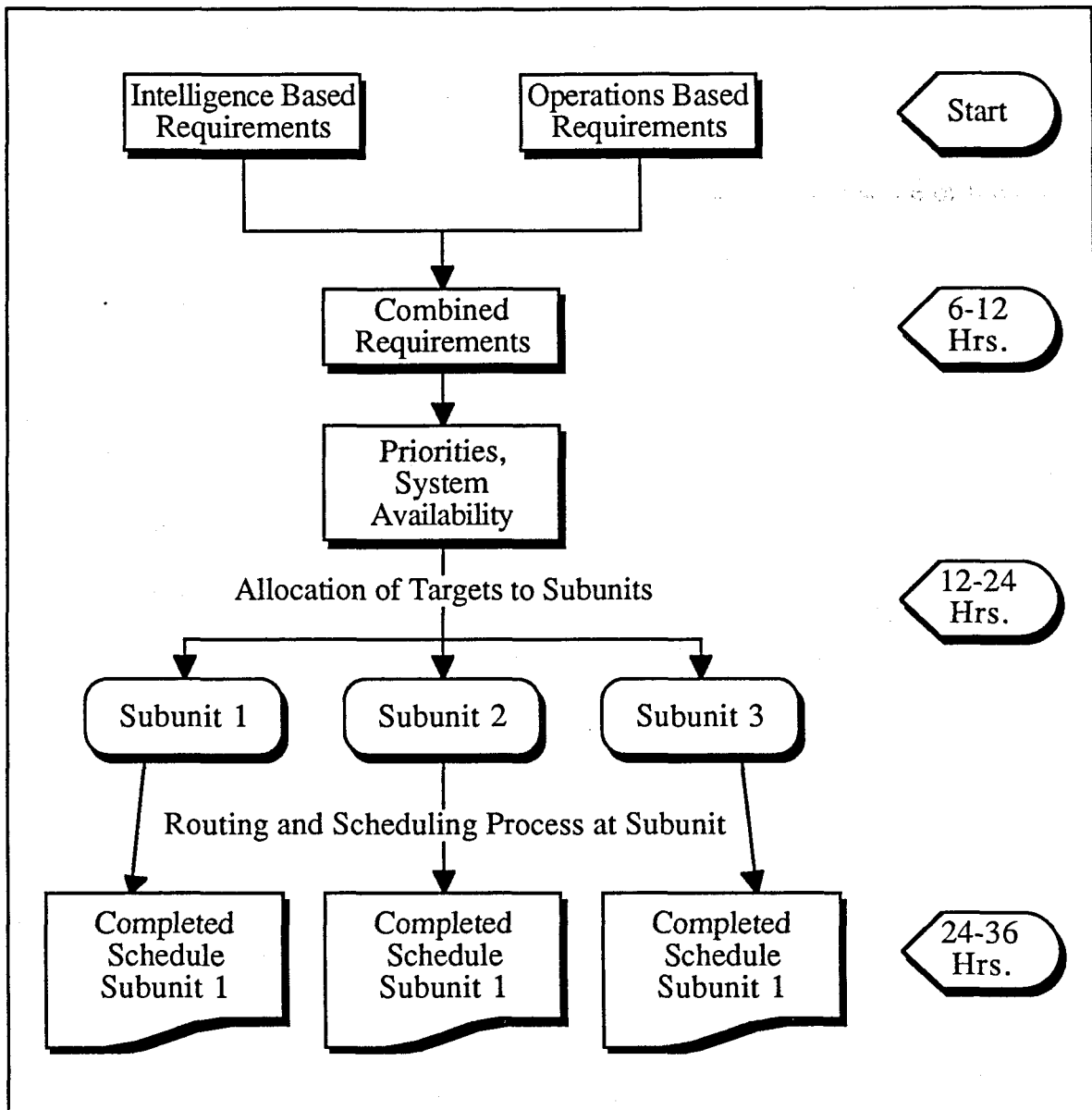
3. Proposed Routing and Scheduling System Improvements

In studying this present system of task development, target assignment, routing and scheduling, and coordination that goes on at various levels of command, it seems to us that certain aspects of the process might be improved, thereby generating benefits both in manpower savings, and in quality of output.

Discussions with Marine Corps TACAIR RECCE and UAV experts [Ref. 2], along with our own observations, bring several areas for improvement to mind. Specifically, we think that the following areas warrant investigation as to possible improvements:

- Selection of targets to visit.

- Assignment of targets selected to appropriate subunits.

- Routing and scheduling of vehicles to visit selected targets.

These aspects are all highly dependent on manual decisions and intervention, and cause problems in the system presently used within the Marine Corps. Successful handling of these points depends greatly on the experience and skill level of the individuals involved in the process, and should therefore be considered candidates for possible automation. Specifically, the selection of targets and their assignment to vehicles is one area that would be able to benefit from the development of a computer algorithm for this task. For the balance of our work, we look for solution techniques and features that would address these points. Our next step, covered in chapter 2, was to review the current literature for any insight that can be gained from previous research.

II. LITERATURE REVIEW

Preliminary work on our problem leads us to believe that some of the aspects of our routing and scheduling problem might be similar to those faced by industry. Therefore, before starting to develop a new model to address our problem, we will review current literature in order to answer several questions, including:

- Has anyone else studied this problem, or one very similar to it?

- If so, have they proposed a solution to this or a very similar problem?

If the answers to the above questions are no, then we are concerned with whether or not any of the recent work lends itself to being adaptable to our problem, or lends some insight to our problem.

A. VEHICLE ROUTING AND SCHEDULING PROBLEM

Consider a situation where a group of customers are to be visited by a fleet of one or more vehicles, with each vehicle performing some service or delivering or picking up some product at each customer. There is a determinable cost to travel between each of the customers and also between each customer and a central depot, from which each vehicle must start and finish. The *Vehicle Routing and Scheduling Problem* (VRSP) is then defined to be the construction of a minimum cost set of vehicle route(s), such that all customer demands are met. Much has been written recently concerning the VRSP problem and its various derivatives. An excellent start on our review of the work is a survey paper on *Vehicle Routing and Scheduling Problems with Time Windows* (VRSPTW) by Solomon and Desrosiers [Ref. 3]. This paper gives a very thorough accounting for the important work in routing and scheduling over the (approximately) last ten years.

The background for the VRSPTW is established by first defining the models for the non-time window constrained problems. The authors cover both optimization and heuristic based approaches to the problem, and include in their article important information concerning research on the various subproblems and related problems within the field. From their work, we are led to investigate the efforts of others.

A thorough treatment of the *Vehicle Routing Problem* (VRP) is given by Christofides et al [Ref. 4:pp. 315-338] in which the authors develop the VRP and then discuss both exact and heuristic solution methods. More recently, Bodin et al [Ref. 5], and Golden et al [Ref. 6], have published work concerning then state-of-the-art approaches to the VRP and extensions. In Bodin et al, the authors present a survey of the different VRP-related problems, along with basic approaches to each. Golden et al show algorithmic methods pertaining to the VRP, models for the VRP, and a short section on practical applications. These works provide valuable awareness of the various problems that were modelled as a VRP or as one of the extensions and generalizations.

From these general treatments of the subject, we turn our attention to some of the more specialized problems in the literature, hoping for insight into our own problem. Some of the papers surveyed include Kolen et al [Ref. 7], Desrosiers et al [Ref. 8], and Solomon [Ref. 9:pp. 254-265]. Kolen et al present a branch-and-bound method for the VRP that minimizes the total route length for a fixed fleet of vehicles. Desrosiers et al give a solution to the VRP that is a column generation technique on a set partitioning problem. Their solution, which uses a linear programming relaxation technique, is very successful in finding integer solutions to the formulation. Solomon's paper, which deals with heuristic solution techniques to the VRSPTW, is an excellent comparison of various tour-building algorithms.

Tour-building algorithms are heuristic methods that attempt to arrive at a solution for the VRP by constructing each vehicle's routing using time, distance, and cost criteria, while using some defined measure to decide which targets to include on a particular vehicle's routing. Solomon shows in his paper a variety of algorithms for developing sequential tours, that is, one vehicle is routed at a time until all customers are visited. This contrasts with the parallel method, in which all vehicles are routed simultaneously. His work was especially thorough in its dealing with time feasibility of customer insertions.

Although we describe our problem as involving routing and scheduling, we do not have any cargo to be delivered or picked up at each of our customers (targets), which contrasts with much of the work done on the VRP. So, any models that incorporate this aspect into the solution might be more involved than necessary for our needs. However, throughout much of the literature on the VRP and extensions, mention is made that this work is closely related to the traveling salesman problem (TSP). Noting this, we next turn our attention to literature concerning the TSP and extensions.

B. TRAVELING SALESMAN PROBLEM

Consider a problem involving a single traveling salesman (vehicle), who is required to visit all of a given number of customers (targets) once. The cost to travel between each of these customers is known. The objective is to find the particular routing that will minimize the total cost of visiting all customers.

The *Traveling Salesman Problem* (TSP) is one of the oldest problems dealing with combinatorial mathematics and optimization. As such, there are hundreds of texts and articles dealing with the formulation and solution of the TSP and its many extensions. One of the best recent texts concerning the TSP is one by Lawler et al

11

[Ref. 10], in which the authors begin at a fairly basic level explaining the motivation and mathematical underpinnings of the TSP, and then proceed to develop many advanced concepts and generalizations for the TSP.

A variant of the TSP is the *Multiple Traveling Salesman Problem* (m-TSP), which attempts to fulfill the requirements of the TSP, but with multiple salesmen (vehicles). The number of salesmen (vehicles) can either be fixed a priori, or allowed to float. This model is an essential portion of many of the vehicle routing problems, and the solution to a m-TSP is considered part of the solution techniques shown in both Kolen et al [Ref. 7], and Desrosiers et al [Ref. 8], among others.

Two additional, more specialized, works studied include Baker [Ref. 11] and Volgenant, Jonker [Ref. 12]. In the former, the author proposes an exact algorithm for the *Traveling Salesman Problem with Time Windows* (TSPTW). Baker concludes that his algorithm is most effective for small (10-30 customers) problems with a relatively large number of customers possessing time windows.

In Volgenant, Jonker, the authors discuss a *Generalized Traveling Salesman Problem* (GTSP) in which not all of the customers are visited, at a penalty cost for not visiting these unrouted customers. The authors show that the GTSP can be transformed into a TSP, and propose techniques for doing so. This paper was important in that it is the first treatment of a TSP or VRP in which the requirement to visit all customers did not exist. In our TACAIR RECCE problem, we feel that this is very likely to be the case. That is, we feel that the potential demand for TACAIR RECCE will outstrip the supply, and that a decision will be need to be made as to which of the available customers (targets) should be included in the final schedule solution.

C. ADDITIONAL RELATED PROBLEMS

Due to its importance as a routing and scheduling model for some problems, some recent work has been directed to finding solution methods for the *Generalized Traveling Salesman Problem* (GTSP). With its relaxation of the mandatory visit requirement, the GTSP is finding numerous applications where customer's demand for service outstrips the supply. This has been treated in a variety of ways by different authors, and some of these techniques will be presented here.

Fischetti, T. and Toth, P. [Ref. 6:pp. 319-344] provide a description of the *Prize Collecting Traveling Salesman Problem* (PCTSP), in which a prize p^* is associated with each potential. The objective in the PCTSP is to find a minimum cost route for the vehicle which collects at least a sum of prizes equal to a goal g, and which visits each customer not more than once. Several models for this problem are proposed by the authors, as well as an exact algorithm for the optimal solution of the PCTSP. Computational experience for a randomly generated problem set is given by the authors.

Closely related to this problem is the *Orienteering Problem* (OP). Its name is derived from the sport of score orienteering, which is a competition in which a competitor travels from a start point to an end point within a fixed period of time, via a set of control points chosen by the competitor from a larger set of control points. Associated with each control point is a score. The object of the game is to maximize one's total score, while not violating the total time constraint. When formulated as a mathematical model, the time factor corresponds to a cost associated with travel from one control point or start/end point to another. The game, therefore, is played as the special case where the cost between points is taken to be the time to travel between these points.

13

Early work on this model was performed by Tsiligirides [Ref. 13], in which he has proposed two heuristics for solving the OP. In one, he uses a randomized process to build a large set of candidate routes, choosing the best of these for his final solution. This procedure uses a measure of "desirability" $A(j)$ on all unrouted customers j, $A(j) = s(j)/t(last, j)$, where $s(j)$ is the score associated with node j, and $t(last, j)$ is the travel time from the last customer routed to customer j. The heuristic then randomly chooses one of the four best of the $A(j)$ values, which corresponds to the next customer j to be routed. This procedure is then repeated until no new customers can be included on the route without violating the maximum time constraint of the OP.

In his other heuristic, Tsiligirides uses a variant of the "cluster first, route second" procedure described in Christofides et al [Ref. 4: pp. 327-334] and Solomon [Ref. 9: pp. 258-264]. Varying the "cluster's" geographic location and size, minimum cost routes are built within each cluster, with the highest score route which does not violate the total time constraint taken as the solution.

Golden, Levy, and Vohra [Ref. 14] propose a heuristic for the OP that combines a center-of-gravity concept with subsequent route improvement. The center-of-gravity idea attempts to use score related information such that the successive routes that are developed by the heuristic are drawn to the score-weighted center of gravity of the points under consideration for insertion. The authors compare their results with those of Tsiligirides [Ref. 13], and present evidence that their procedure performs at least as well as Tsiligirides' heuristics in most cases, and significantly outperforms his heuristics in others.

Golden,Wang and Liu [Ref. 15], in a follow-on to Golden et al's [Ref. 14] work, propose enhancements to the previously developed heuristic described

14

above. The two most important of these enhancements are given here. In the first, a concept of *subgravity* is added to the idea of center-of-gravity that was earlier introduced. Subgravity is the authors' term for a method that allows for determining clusters of control points, and the subsequent ability to obtain multiple scores cheaply. A second new idea is one of "learning", in which the ability of the heuristic to effectively determine which combinations of control points tend to be included on prospective routes with the highest scores is determined, with this information then subsequently used within the heuristic.

Finally, an optimization based approach to the OP has been proposed by Ramesh, Yoon, and Karwan [Ref. 16]. Their method involves optimal algorithms using problem reformulation and Lagrangean relaxation. The authors report results for problems of up to 150 control points, and suggest future directions for optimization based research.

D. LITERATURE REVIEW CONCLUSIONS

Though much work has been done recently concerning the VRP, TSP and related problems, we feel that no one so far has quite captured the exact model of our problem. However, we do feel that there is a possibility that by combining the ideas contained within several of the works cited, that an adequate model can then be developed for further study. Specifically, the time-window constraints treatment of Solomon [Ref. 9: pp. 254-265], the set partitioning formulations of Christofides [Ref. 4:pp. 315-338], and Desrosiers, Soumis, Desrochers [Ref. 8], and the heuristics developments of Tsiligirides [Ref. 13] and Golden, Levy, Vohra [Ref. 14] and Golden,Wang, Liu [Ref. 15] all deal with what could be portions of a potential model for our problem. It is this model that we will present in the next chapter.

III. SOLUTION METHODOLOGY

In this chapter we examine the solution methods used for our TACAIR RECCE vehicle routing and scheduling problem. We begin this by stating a mathematical program for our problem. Next we examine the solution approaches attempted, and discuss the models and algorithms included in these proposed solution methods.

A. MODEL DEVELOPMENT

In attempting to find a solution to almost any type of problem that seems to have the potential to be approached from an optimization standpoint, it helps to first develop a mathematical program (MP) formulation of the physical situation. Since the possibility for optimization based solutions seems to exist for our problem, we set out to formulate the problem as an MP.

Our routing and scheduling problem for TACAIR RECCE vehicles has the following characteristics. Let N be a set of targets numbered $i = 1,2,...,NUMTGT$, which includes a depot which is target number 1. The nonnegative cost c_{ij} to travel between each pair of targets (i,j) is the distance between these targets, although it could represent some other type of determinable cost. Associated with each target i is a nonnegative priority point value p_i, except for the depot which is assigned $p_1 = 0$. Also, each target has a service time, s_i, which is the time required for a vehicle to remain at a target so as to service the target. The travel costs between targets can vary between vehicles, but the service time is the same regardless of which vehicle serves the target. Additionally, each target has a time window defined by the earliest time to begin service, called e_i, and the latest time by which service must be started, called l_i. The decision variables associated with the model, therefore, are the times that a

16

vehicle arrives at target i, given by a_i, and the time that service at target i begins, given by b_i. The wait time at target i, w_i, is calculated as $w_i = b_i - a_i$.

Let M be a set of vehicles numbered $m = 1, 2, ... NUMVEH$, for which there exists a maximum airborne time for each given by $VTMAX_m$. Let x_{ijm} be a decision variable that will take on the value 1 if vehicle m travels from target i to target j, and 0 otherwise. The mathematical programming formulation of the problem is given below. Due to the problem's similarity to both the orienteering problem and to the multiple traveling salesman problem with time windows, we decided to call this problem the *Multi-Player Orienteering Problem with Time Windows* (MPOPTW).

<u>Multi-player Orienteering Problem with Time Windows (MPOPTW)</u>

MP: Maximize $\displaystyle\sum_{j \in N} p_j \sum_{m \in M} \sum_{i \in N} x_{ijm}$ (3.1)

$s.t.$ $\displaystyle\sum_{m \in M} \sum_{i \in N} x_{ijm} \leq \begin{cases} NUMVEH & \text{if } j = 1 \\ 1 & \text{if } j = 2,3,...,n \end{cases}$ $\forall j \in N$ (3.2)

$\displaystyle\sum_{m \in M} \sum_{j \in N} x_{ijm} \leq \begin{cases} NUMVEH & \text{if } i = 1 \\ 1 & \text{if } i = 2,3,...,n \end{cases}$ $\forall i \in N$ (3.3)

$\displaystyle\sum_{i \in N} x_{ijm} - \sum_{i \in N} x_{jim} = 0$ $\forall j \in N, \forall m \in M$ (3.4)

$\displaystyle\sum_{i \in N} \sum_{j \in N} c_{ijm} x_{ijm} \leq VTMAX_m$ $\forall m \in M$ (3.5)

$\displaystyle\sum_{i \in S} \sum_{j \notin S} x_{ijm} \geq y_s$ $\forall S \subseteq N, \{1\} \subset S, \forall m \in M$ (3.6)

$\displaystyle\sum_{i \notin S} \sum_{j \in S} x_{ijm} \leq y_s (|N - S|)$ $\forall S \subseteq N, \{1\} \subset S, \forall m \in M$ (3.7)

$\displaystyle\sum_{m \in M} x_{ijm} \leq 1$ $\forall i, j \in N$ (3.8)

$y_s \in \{0,1\}$ $\forall S \subseteq N, \{1\} \subset S$ (3.9)

$$x_{ijm} \in \{0,1\} \qquad \forall i,j \in N, \forall m \in M \qquad (3.10)$$

$$b_i + c_{ij} \le b_j \qquad \forall \{i,j \,|\, x_{ijm} = 1\} \qquad (3.11)$$

$$e_i \le b_i \le l_i \qquad \forall \{i \,|\, x_{ijm} = 1\} \qquad (3.12)$$

Constraints (3.2) and (3.3) ensure that the total number of vehicles leaving the depot is not exceeded, and that the maximum number of vehicles visiting each of the other targets is not greater than one. In addition, when taken with constraint (3.4), these constraints ensure that the number of vehicles entering a target equals the number of vehicles leaving a target. Constraint (3.5) ensures that the maximum airborne time for any vehicle is not exceeded. Constraints (3.6) and (3.7) are used to eliminate subtours, by the use of the target set partitioning variable y_S and the partition set S. Constraint (3.8) ensures that each target is visited by at most one vehicle. Equations (3.9) and (3.10) set the domain for the formulation variables. Finally, the time window restrictions on the targets are enforced with logical constraints (3.11) and (3.12). This particular approach is similar to formulations given in Bodin et al [Ref. 5: pp. 83-90] and Ramesh, Yoon, and Karwan [Ref. 16; pp.4-5].

Inspection of this formulation shows several features which make it difficult to both implement and solve. First, as a mixed integer formulation, the practical size of any problem that could be solved using currently available software is limited, especially since the integer variables are indexed on three different sets. Additionally, implementation of several of the constraints would be challenging due to the various conditionals on their execution. For instance, the logical constraints (3.11) and (3.12) require a large number of additional binary variables in order to be recast as

true mathematical programming constraints [Ref. 17:pp. 186-196]. With these thoughts in mind, it is easily seen that this formulation is too formidable to implement directly, so we will not use it as shown.

In the next two sections, we describe two different solution approaches that were attempted. The first is an optimization approach based on a set partitioning model. The second is a heuristic solution approach that we developed for our problem. The heuristic approach is the one selected for full implementation for the TACAIR RECCE problem.

B. OPTIMIZATION BASED SOLUTION APPROACH

Consider a model for our problem that solves a set packing problem, with columns corresponding to feasible vehicle routes and rows corresponding to individual targets. A vehicle route is defined as a routing of a single vehicle over a subset of targets which is feasible with respect to target time windows and total route length. The objective for the model is to maximize the sum of the priorities of the targets routed, subject to an upper limit on the total number of vehicles available and the fact that each target may be scheduled only once. This model is similar to one involving set partitioning, used on a vehicle routing problem, and proposed by Desrosiers, Soumis, and Desrochers [Ref. 8]. We chose to explore a model similar to theirs so as to decide if any benefits might come from this approach. This model can be viewed as a dual transformation of their model, since we are maximizing priority points subject to time window and total cost constraints, while their model minimizes cost subject to meeting every target, while meeting time window requirements. The reason for our transformation to a set packing formulation goes back to the basic idea of the MPOPTW, that is, that not all targets will be routed. Set

19

packing captures these ideas. A simplified representation of this model is given below using the same notation as before with the following addition.

Let R be the set of feasible routes generated for input to the constraint matrix, numbered $r = 1,2,...NROUTES$. Then let d_{ri} take on value 1 if target i is visited on route r, and 0 otherwise. Our decision variable is $t_r = 1$ if route r is selected for inclusion into the final solution, and 0 otherwise.

$$\text{Maximize} \qquad \sum_{i \in N} p_i \sum_{r \in R} d_{ri} t_r \qquad\qquad (3.13)$$

$$s.t. \qquad \sum_{r \in R} d_{ri} t_r \leq 1 \qquad\qquad \forall i \in N \quad (3.14)$$

$$\sum_{r \in R} t_r \leq NUMVEH \qquad\qquad (3.15)$$

$$t_r \in \{0,1\} \qquad\qquad \forall r \in R, \forall i \in N \quad (3.16)$$

The notable feature of this model is the need to generate the set R of possible routes, which in practice may be astronomically large. Desrosiers et al chose to generate routes via a shortest path algorithm with time windows on the nodes (targets). In our work, we developed our sets of feasible routes manually, and used these for input to our solver software.

C. HEURISTIC BASED SOLUTION APPROACH

The orienteering problem has been shown to be NP hard by Golden, Levy, and Vohra [Ref. 14]. By reduction, our problem is also NP hard since we could set $NUMVEH = 1$ and all time windows infinitely wide. Thus, heuristic methods should be considered for the problem, since these give the most potential for solving problems of realistic size.

Our heuristic attempts to find the set of feasible routes (hereafter called a *schedule*) that maximizes the sum of the priorities for the targets included in the

schedule. In doing so, we expand upon the work of previous authors and combine their notions with our own ideas to arrive at a heuristic solution to the MPOPTW. The main idea behind the heuristic is that of developing a sequence of *candidate schedules*, using a randomized route initialization process in combination with a defined target selection and insertion technique, and keeping the best of these candidate schedules as the final solution. Each candidate schedule consists of a set of routes, built sequentially and referred to when incomplete as an *emerging route*. In each of the next sections, we describe the major components of our heuristic. A list of variables used in the algorithm descriptions is given below.

GRAND	Sum of priority points of targets scheduled on incumbent schedule.
NUMRUNS	Maximum number of candidate schedules developed.
TOTPTS	Sum of priority points of targets scheduled on current candidate schedule.
COST(i)	Cost of inserting target i on emerging route.
VEH	Vehicle currently being routed.
NUMTGT	Number of targets to be considered for routing.
NUMVEH	Maximum number of vehicles available.
TGTSKD	Number of targets scheduled so far on current candidate schedule.
INITGT	Target used to initialize a route. Can be either random based selection or forced.
NOSPOT	Logical flag denoting that no feasible target insertions remain for the emerging route.

DOALL	Logical flag denoting the first pass through the target list during an emerging route's development. Used to reduce computation time, by taking advantage of previously gathered information concerning target time feasibilities.
ICHOIC	Target chosen to be inserted next into the emerging route.
CURBST	The current best target to be inserted into the emerging route. Used to help determine *ICHOIC*.
ISPOT(i)	Vector returned from **BSTSPT** denoting the currently scheduled target *after* which the target *i* should be inserted in the emerging route.
FIRST(VEH)	The target which a particular vehicle will be forced initialized, if any.
VSCORE(VEH)	The sum of the priority points for a particular vehicle *VEH*.
TGTVEH(i)	The vehicle to which target *i* has been assigned within the current candidate schedule.

1. Main Control Algorithm (MAIN)

The main control algorithm of the heuristic involves establishment of data structures, determination and control of the number of candidate schedules to be examined, comparison between the incumbent schedule and each successive candidate schedule, and the output of the final incumbent schedule as the solution. This algorithm is shown in pseudocode form in Figure 3.

```
Input:    Target and vehicle input data structures
Output    Listing, by vehicle, of routes for solution schedule

Do BEGIN →
   Establish input and output files
   GRAND = 0
   Do (i = 1 to NUMRUNS) →
      Determine candidate schedule /* call CANSCED */
      Compute TOTPTS = sum of priorities for routed targets on candidate
            schedule
      If (TOTPTS > GRAND) →
         GRAND = TOTPTS
         Store candidate route as new incumbent
      fi
   od
   Output incumbent schedule as solution
od
```

Figure 3. Main Control Algorithm (MAIN)

In order to more effectively implement this system, we separated the major areas of the algorithm into subroutines. These are listed below for reference, and are described in detail in the following sections.

CANSCED   Candidate schedule development

ROUTINIT   Route initialization

BSTSPT   Determination of best insertion spot for each unrouted target

INSERT   Target insertion on emerging or improving route

IMPROV   Candidate schedule improvement routine

2. Candidate Schedule Development Algorithm (CANSCED)

This algorithm is the heart of the heuristic, as it develops each of the candidate schedules used to determine the final solution. It computes each candidate

schedule by sequentially developing *NUMVEH* different routes, each corresponding to a particular vehicle. In developing each route, **CANSCED** uses a randomized route initialization process (**ROUTINIT**), target insertion algorithms (**BSTSPT** and **INSERT**), and finally a candidate schedule improvement algorithm (**IMPROV**). Also, within **CANSCED** itself, the next target to be inserted into an emerging route is determined by a greedy heuristic which takes as the next insertion that target which has the highest value of $p(j)/COST(j)$, where $COST(j)$ is given as the additional cost of inserting target j on the emerging route.

In order to determine the candidate schedule, **CANSCED** works "forward" through the list of vehicles, completing all possible target insertions on each vehicle before proceeding to the next vehicle. This is repeated for each vehicle until no more targets can be routed for any vehicle. Then, an attempt is made to improve the candidate schedule. Once this has occurred, control is returned to MAIN to determine whether this latest candidate schedule should be retained as the new incumbent or discarded. This algorithm is given is Figure 4.

3. Route Initialization Algorithm (ROUTINIT)

Route initialization can be accomplished by either of two means. The first is through use of a randomized process, where the route is initialized with one of the k highest priority targets still remaining unscheduled on the emerging candidate schedule. This value of k can be altered to allow for parameterization of the process. This partially randomized process is similar to one used by Tsiligirides [Ref. 13], although he used this procedure not only for route initialization but also for selection of each target to be inserted on the emerging route.

24

```
Input:     Target & vehicle data structures
Output:    Return to main control algorithm with candidate schedule

Do Begin →
    VEH = 1
    TGTSKD = 0
❶   Do while ((TGTSKD < NUMTGT) & (VEH < NUMVEH)) →
        Initialize emerging route using randomized process, returning the target
            number selected for initialization as INITGT  /* call ROUTINIT */
        If (INITGT = 0) →
            Print ("All targets scheduled with "VEH" total vehicles")
            Exit Do while ❶
        fi
        NOSPOT = .FALSE.
        DOALL = .TRUE.
❷       Do while (TGTSKD < NUMTGT)
            Determine best spot for insertion into emerging route for each
                unrouted target  Return array of additional cost values c(j) and best
                insertion spots ISPOT(j) from BSTSPT. If no insertions feasible,
                return NOSPOT = .FALSE.   /* Call BSTSPT */
            If (NOSPOT = .TRUE.) → Exit Do while ❷   fi
            DOALL = .FALSE.; ICHOIC = 0; CURBST = 0
            For (j = 1 to NUMTGT) →
                If (p(j)/COST(j) > CURBST) → CURBST = p(j)/COST(j);
                    ICHOIC = j;
                fi
            rof
            Insert ICHOIC into the emerging route /* Call INSERT */
            TGTSKD = TGTSKD + 1
        od
        VEH = VEH + 1
    od
    If nonstandard exit from loops → Print error message
    Call candidate schedule improvement routine  /* call IMPROV */
od
```

Figure 4. Candidate Schedule Development Algorithm (CANSCED)

In addition to the procedure described above, each route can be "forced
initialized" to a particular target, with this selection given as FIRST(VEH) in the
input. This corresponds to manually overriding the heuristic so as to guarantee that

25

a particular target is routed, or to ensure that a particular vehicle is used to visit the target.

In either of the cases stated above, the target is scheduled so as to take advantage of as much future flexibility as possible. Therefore, the vehicle is scheduled to leave the depot in order to arrive at the target and begin service at a time that places the service in the middle of the target's time window. This allows for the maximum target scheduling movement and therefore flexibility of the emerging route's depot departure time. The pseudocode for the algorithm is given below in Figure 5.

```
Input:    Target and vehicle data structures
          VEH /* vehicle currently being routed */
          FIRST(VEH) /* "forced" initialization target, equals 0 if none */
Output:   Initialized emerging route, with updated data structures

Do Begin→
    If (FIRST(VEH) = 0) →
        INITGT = One of k highest priority targets randomly selected for route
            initialization
    Else →
        INITGT = FIRST(VEH)
    fi
    Initialize vehicle_veh with INITGT and update data structures
    p(j) = - p(j) /* set routed target's priority to negative for flag purposes */
    TGTVEH(INITGT) = VEH
od
```

Figure 5. Route Initialization Algorithm (ROUTINIT)

4. Best Insertion Spot Algorithm (BSTSPT)

In order to determine which of the unrouted targets should be next inserted into each emerging route, we need to determine where the most economical

point is to insert each of these unrouted targets. To determine this, we calculate the additional cost, in terms of lengthened emerging route times, for each unrouted target's possible insertion points on the emerging route. In doing so, we use a procedure that is an extension of Solomon's [Ref. 9: pp. 255-256] and Prof. R. E. Rosenthal's (as referenced by Chun and Lee [Ref. 18:pp. 43-46]) procedures. In his work, Solomon proves a lemma for determining the necessary and sufficient time feasibility conditions for inserting a customer on a partially constructed route. He utilizes a concept of *push forward* in the schedule at customer i, which he defined as the difference between the original service begin time at a customer before any insertion, and the new begin service time after an insertion. The push forward at each target subsequent to an insertion into an emerging route is defined as

$$\text{push forward}_{i+1} = \max\{0, \text{push forward}_i - w_{i+1}\} \qquad (3.17)$$

Solomon states that a concept of *push backwards* can be similarly defined, but that to do so is not appealing since a schedule can be pushed backwards only if the vehicle leaves the depot at some time after the earliest possible departure time. For our problem, however, it is very likely that some vehicles may in fact leave the depot at a time after the earliest possible, since our vehicles are limited by $VTMAX_m$ and may have to wait for time windows to open. (See the description of **ROUTINIT** for more details on vehicle departure time determination). Therefore, to assist in determining time feasibility of insertions, we define the following values for each target p on the emerging route (note start and endpoints are the depot, $p = 0$), and for each vehicle $m = 1,2,... NUMVEH$. Additionally, let S_m be the set of all targets visited by vehicle m.

$$MAXPF_p = \min\{MAXPF_{p+1}, l_p - a_p\} \qquad (3.18)$$

27

$$MAXPB_{p+1} = \min\{MAXPB_p, b_p - e_p\} \tag{3.19}$$

$$SLIDEF_m = \min\{l_p - a_p \quad \text{for } p \in S_m\} \tag{3.20}$$

$$SLIDEB_m = \min\{b_p - e_p \quad \text{for } p \in S_m\} \tag{3.21}$$

The value $MAXPF_p$ measures the amount by which a target and all others after it can be *pushed forward* in time while not causing time feasibility problems for any previously scheduled target. By including the necessary computations, the total time factor $VTMAX_m$ is included within this concept, by establishing changing e_p and l_p values for the departure and arrival depot nodes. In a similar manner, $MAXPB_p$ is used to determine time feasibility constraints for any target p and all others *before* it in an emerging route, by determining a maximum *push backward* at p that is time feasible. By their nature, the concept of $MAXPF_p$ and $MAXPB_p$ entail an alteration in the total length (and therefore, cost) of an emerging route. Note that the computation of $MAXPF_p$ and $MAXPB_p$ is recursive and can be done very quickly.

The values $SLIDEF_m$ and $SLIDEB_m$ measure the amount by which *every* target on an emerging route can *slide* its arrival and begin service time moved either forward (in the case of $SLIDEF_m$) or backward (for $SLIDEB_m$) and not change the relative time relationships and therefore the time feasibility conditions already established. The $SLIDEF_m$ and $SLIDEB_m$ values do not affect any previously determined total route length values.

Within the heuristic, these values are used in combinations and in conjunction with the wait times, w_i, to determine and test for time feasibility of an insertion into the emerging route. In particular, many times a combination of these *pushes* and *slides* produce the most economical insertion for a certain target/potential

28

insertion point test pair. The complete algorithm for determining the best insertion spot is given in Figure 6.

5. Target Insertion Algorithm (INSERT)

Once the target to be inserted on the emerging route has been selected within **CANSCED** and designated as *ICHOIC*, this information is passed to the **INSERT** algorithm for execution. This algorithm adjusts all data structures, and resets any previously set flags, so that when control is returned to **CANSCED** the data structures are ready for continuation of the sequential route building process. The insertion algorithm is given in Figure 7.

6. Candidate Schedule Improvement Algorithm (IMPROV)

Once all vehicles have been scheduled with the maximum number of targets possible in accordance with the heuristic, an attempt is made to move some of the targets from one vehicle to another, thereby freeing space into which additional, previously unscheduled targets might be inserted. This is accomplished within our heuristic by moving *backward* through the set of vehicles, at each step allowing the vehicle to schedule any target, previously unscheduled or previously assigned to another vehicle with a smaller vehicle index subscript. If the target was previously routed, the target is removed from its former vehicle assignment and is reassigned to the new vehicle. This will create the possibility for additional insertions into the old vehicle when the algorithm reaches the point at which it is examining the old vehicle for possible insertions. Note that the total priority score for the candidate schedule will only increase if a target that was unrouted prior to entry into **IMPROV** is routed at the end of the algorithm, and this will occur only if some movement of previously *scheduled* targets *between* vehicles is accomplished. The algorithm for this is given in Figure 8.

29

```
Input:      Target & vehicle data structures
            DOALL /* Logical for determining if all targets need to be tested */
Output:     ISPOT() /* previously routed target following which examined
                       target should be inserted for lowest additional cost */
            COST(j) /* total additional insertion cost for each unrouted target j */
            NOSPOT   /* = .TRUE. if no targets can be feasibly inserted
                        = .FALSE. otherwise */


Do Begin →
   NOSPOT = .TRUE.
❶    For (j = 1 to NUMTGT) →
        If ((DOALL = .FALSE.) & (c(j) = ∞)) → Continue for ❶;  fi
        c(j) = ∞
        ISPOT(j) = 0
        If (p(j) < 0) →  Continue for ❶;  fi
❷       For (each previously scheduled node on the current route i, except last
              depot node) →
           Compute c_{ij}, c_{j,i+1}, c_{i,i+1}, and temporary route parameters
           If (arrival at j is feasible) → /* test using SLIDEF, SLIDEB, MAXPF,
                                              MAXPB values */
              If (arrival at i + 1 is feasible) → /* test using SLIDEF, SLIDEB,
                                                     MAXPF, MAXPB values */
                 If (total insertion cost <c(j)) & (total vehicle cost is not violated))→
                    c(j) = total additional insertion cost
                    ISPOT(j) = i
                    NOSPOT = .FALSE.
                 fi
              Else →
                 c(j) = ∞
                 Continue for ❷
              fi
           Else →
              c(j) = ∞
              Continue for ❷
           fi
        rof
     rof
od
```

Figure 6.   Best Insertion Spot Algorithm (BSTSPT)

```
Input:     Target and vehicle data structures
           ICHOIC /* target to be inserted into emerging route */
           ISPOT(j) /* previously routed target following which examined
                      target j should be inserted for lowest additional cost */
           j /* index of target to be inserted on emerging route */
Output:    Revised target and vehicle data structures after ICHOIC is inserted
           into emerging route

Do Begin→
    i = ISPOT(j)
    i + 1 = next previously scheduled target on route
    Determine if insertion of j between i and i + 1 causes route error
    If error → Print error message and return; fi
    VSCORE(VEH) = VSCORE(VEH) + p(j)
    p(j) = - p(j) /* set priority of routed target to negative for flag purposes */
    TGTVEH(INITGT) = VEH
    For (each node p on route) →
        Compute MAXPF_p , MAXPB_p , a_i, b_i
    rof
    Compute SLIDEF, SLIDEB, and launch time for route
od
```

Figure 7. Target Insertion Algorithm (INSERT)

31

```
Input:     Candidate schedule with associated data structures
           NUMVEH /* total number of vehicles */
Output:    Improved or unchanged candidate schedule

Do Begin →
   For (all previously scheduled targets j) →
      p(j) = abs(p(j)) /* used for flag purposes */
   rof
❶  For (VEH = NUMVEH to 1 by -1)
      NOSPOT = .FALSE.
      DOALL = .TRUE.
      Do while (continue criteria not satisfied) →
         Determine ISPOT(i) and COST(j) for each unrouted target i
            /* Call BSTSPT */
         If (NOSPOT = .TRUE.) →
            For (ii = 1 to NUMTGT) →
               If (TGTVEH(ii) = (VEH -1)) →
                  p(ii) = - abs(p(ii))  /* used for flag purposes */
               fi
            rof
            Continue For ❶
         fi
         DOALL = .FALSE.
         j = argmin (COST(j))
         If (j was previously routed on another vehicle) →
            Remove j from old vehicle
            Adjust old vehicle data structure and update parameters
         fi
         INSERT j into vehicle_VEH at ISPOT(j)
         Update vehicle_VEH data structure and parameters
      od
   rof
od
```

Figure 8.   Candidate Schedule Improvement Algorithm (IMPROV)

IV. COMPUTATIONAL STUDY

In this chapter we describe the implementation of our solution methods. We also show the development of the problem sets that we use for the test input, along with graphically portraying both the input data and the output solution.

A. OPTIMIZATION BASED SOLUTION

In order to gauge the effectiveness of our small optimization based model (see Chapter III, Section B), we implemented the set packing model using LINDO, a general purpose linear and integer programming software package available on various computing platforms [Ref. 17]. A small data set consisting of ten targets and two vehicles was used to evaluate this model, with the columns which correspond to different feasible routes also developed manually.

The results of this experiment were mixed. Using LINDO's integer programming capability to solve the set packing model produces a solution which provides an optimal answer as to which of the routes should be flown. The results of solving the model as an LP, however, are not as encouraging. Fractionation of the resulting output is severe enough to discount any capability of the results being easily converted to integers, which is necessary to build a complete schedule. Therefore, no benefit could be realized in using the LP solution technique over using the IP model. A change of formulation or solution approach might produce better results, and indeed in Desrosiers et al [Ref. 8] they describe a solution technique for a similar model that uses a branch-and-bound technique to eliminate this fractionation.

However, as noted earlier in our discussion of the optimization model development, column generation for the input is still the main problem with this approach. Developing the columns, which correspond to feasible routes for the vehicles, is tackled as an m-TSP problem by Desrosiers et al [Ref. 8], and the routes so developed are used as input to the branch-and-bound solution technique mentioned above. However, developing the feasible routes is quite a substantial task, and for any large number of targets, would be computationally expensive. In fact, as the time windows become larger, and therefore less of a constraining factor, the enumeration of all possible routes becomes combinatorially explosive, as it approximates total enumeration of a TSP. In light of this, we next describe our implementation of the heuristic model described in Chapter 3.

B. HEURISTIC BASED SOLUTION

To test our heuristic model, we programmed our algorithms using FORTRAN on an Apple Macintosh II computer. Standard FORTRAN 77 protocol is used, in order to make the code more transportable between platforms. Data set development is with Wingz, a Macintosh based spreadsheet. A description of these follow.

1. Data Set Development

We generated various size data sets for testing the heuristic, with help from Marine Corps TACAIR RECCE experts in setting the various input parameters for the data. We also generated data for problems that were much larger than any problem the Marine Corps has ever attempted to solve with their current manual methods.[3] Input was gathered on typical target time window characteristics, along

---

[3] Problems of such large size will be relevant if the Marine Corps centralizes the target allocation decisions as suggested in this thesis.

with rough guidance concerning vehicle airborne time and range. The specifics of this data set development, along with the test data set used for our example, are shown in Appendix A.

2. Computational Results

Using the data sets developed above, we then ran our FORTRAN code of the algorithms shown in Chapter 3. This code was developed with the capability for output of intermediate results, along with a continuous output of the status of the route building algorithms involved. After the total number of candidate schedules (*NUMRUN* in the algorithm descriptions) has been developed and compared, the best schedule is printed and stored to disk as the solution. This solution is in the form of a simplified flight schedule for the vehicles involved. An abbreviated sample of the intermediate output and the final solution output for our test problem is given in Appendix B. It is of interest to note that the best candidate schedule total priority point score (*GRAND* in the algorithm descriptions) improved from the first run value of 4129 points to an overall best (and final output) value of 4659 points, an increase during the 50 runs of candidate schedule development equalling approximately 13 percent. In addition to the data set shown here for example, we also tested the heuristic on data sets ranging from 75-175 customers (targets). Results similar to those of the example data set were obtained.

On our programming platform, an Apple Macintosh II computer, we adjusted the value of *NUMRUN* and compared the run times for the 150 target data set. For the example data set, the computation of 50 candidate schedules (*NUMRUN* = 50) took 1491 seconds, although this number can be reduced substantially by changing the manner in which data is read in to the program, and suppressing the intermediate output. Testing reveals that this time could be reduced

by over 50 percent through these two steps. With this as a guide, approximate run times of 11 - 13 minutes for 50 candidate schedules, using 150 targets and 7 vehicles, seem within reach for this particular platform.

The target location data for our test problem along with the heuristically computed vehicle routings are shown graphically in Figures 9, 10, and 11. Of note is the fact that the routes tend to "cross back" on themselves, which would be suboptimal for the TSP without time-window constraints. However, for our time-window constrained targets, it is extremely likely that many such crossings will occur, since the temporal aspects will many times force the scheduling factors to outweigh the routing factors. Also, note that the launch point for Vehicle 1 and the drop-off point for Vehicle 2 are distinct from the depot, and are the result of input to that effect for a "forced initialization" of targets for these respective vehicles.
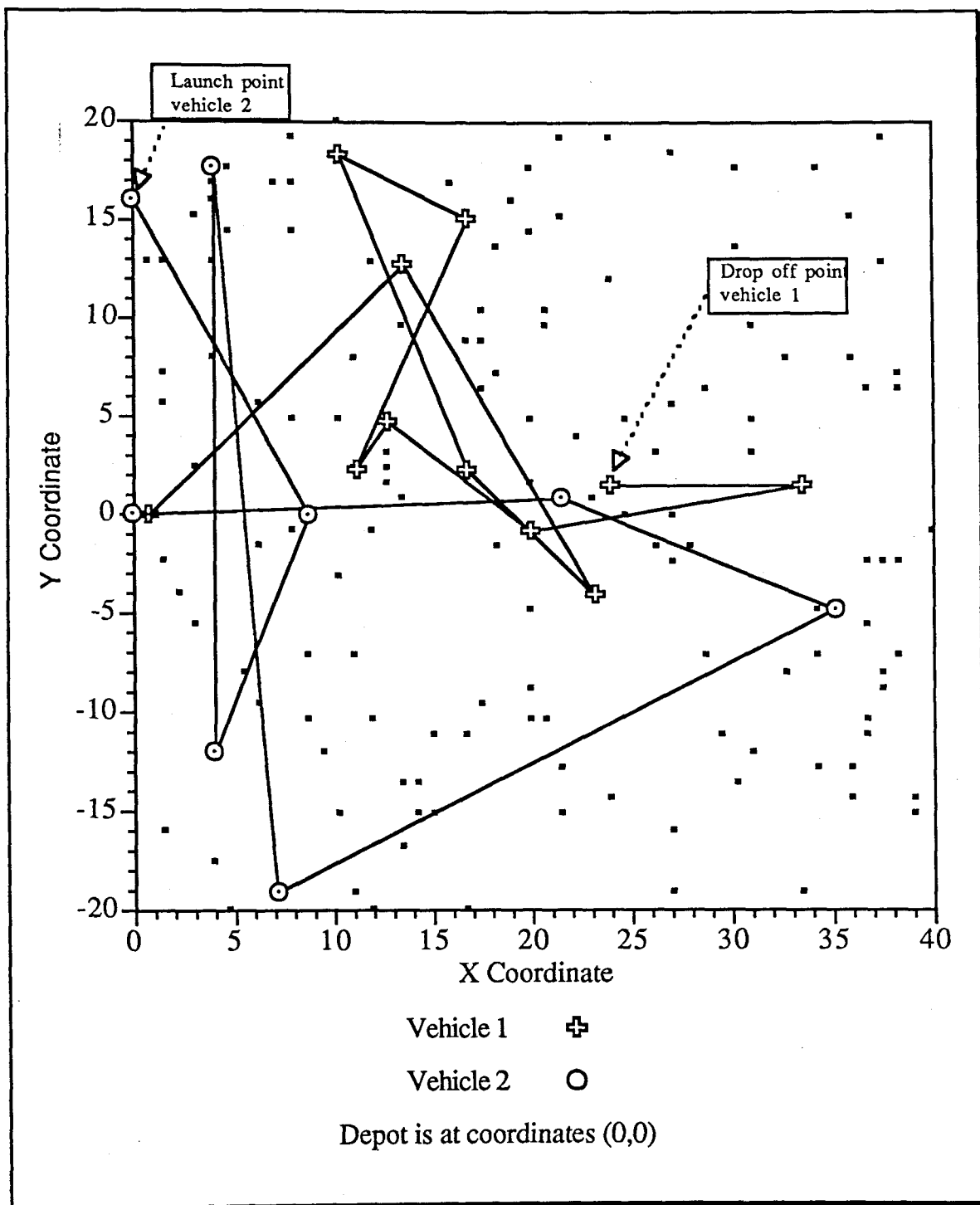
Figure 9. Example Problem Final Schedule Vehicles 1, 2
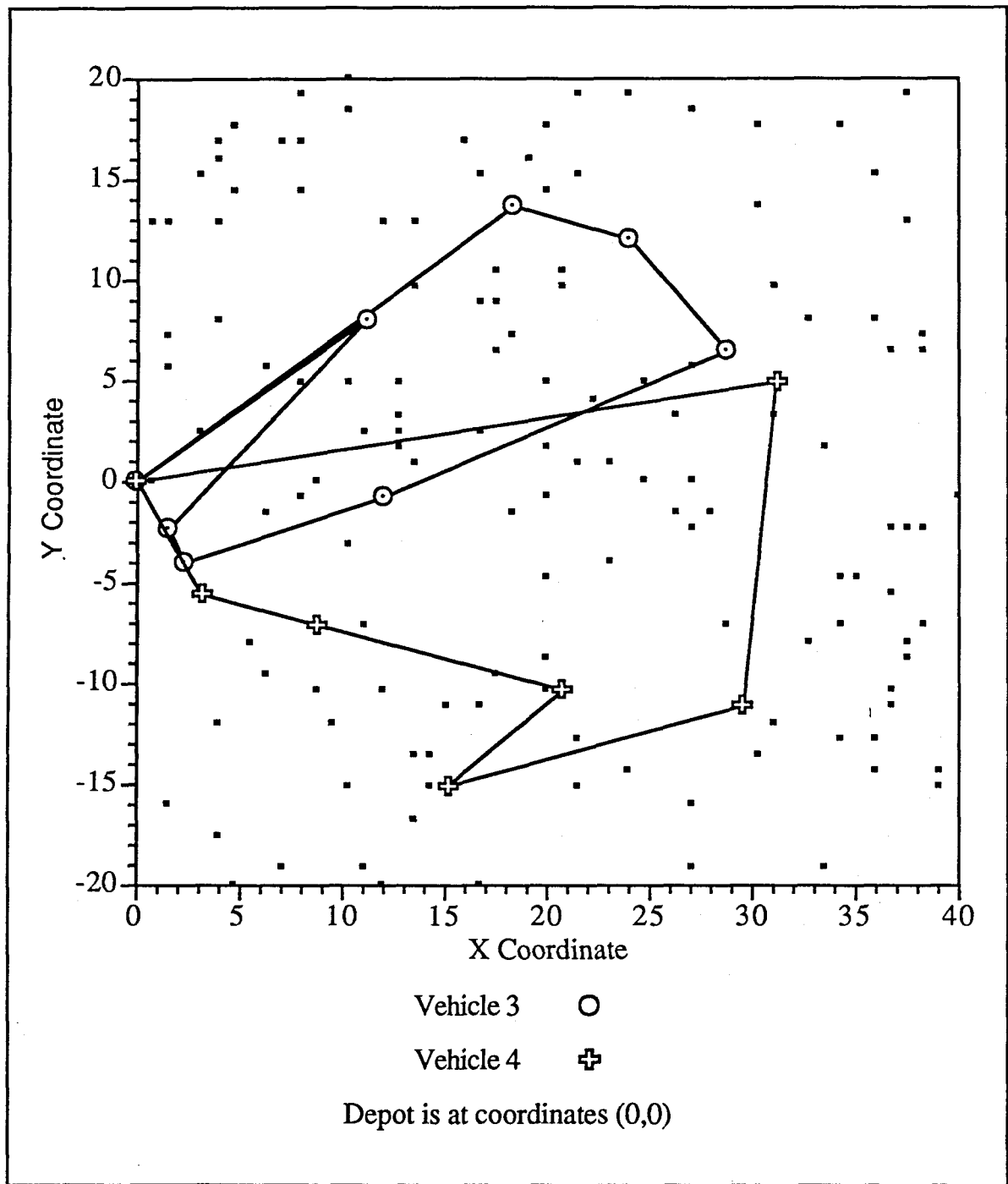
Figure 10. Example Problem Final Schedule Vehicles 3, 4
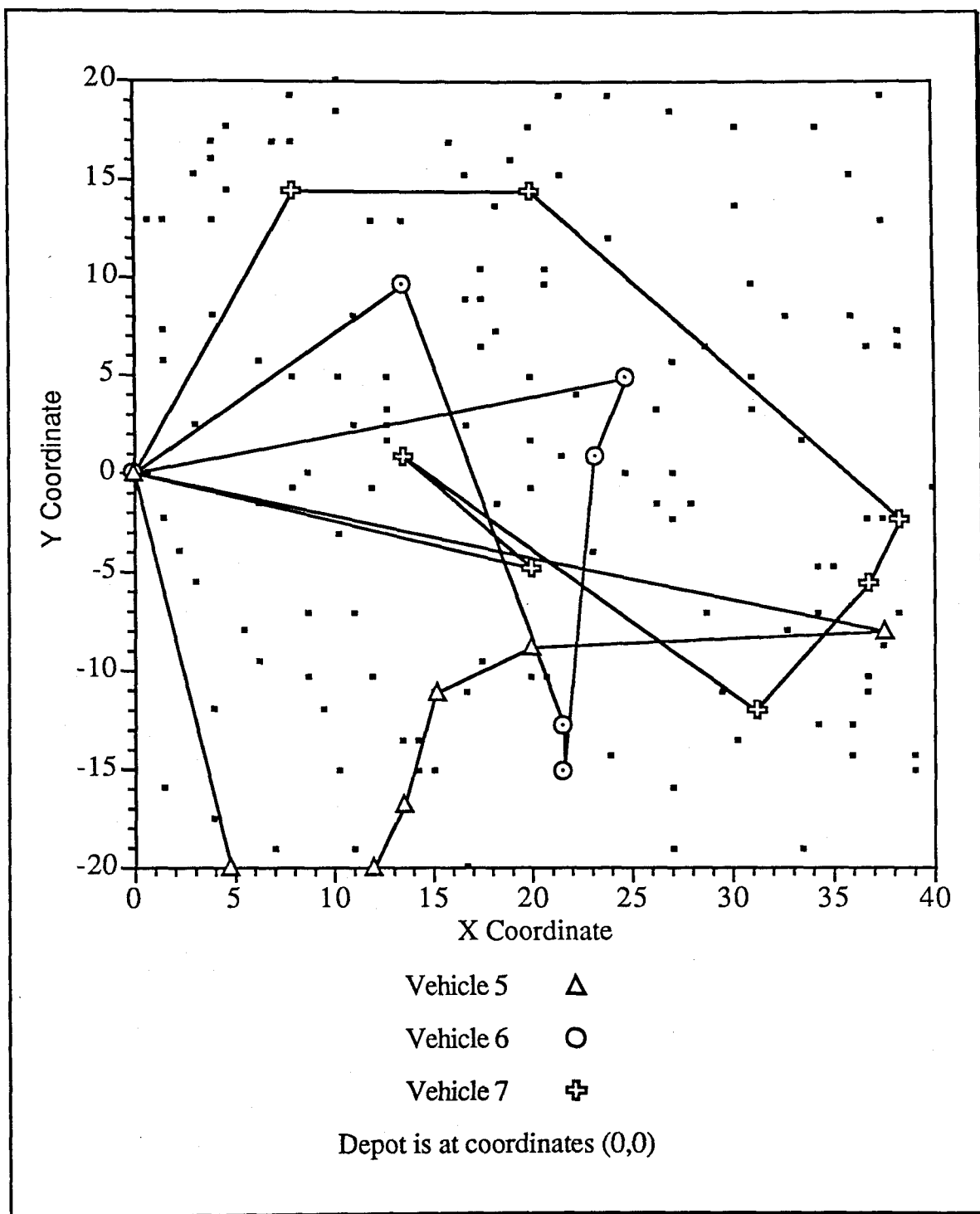
Figure 11. Example Problem Final Schedule Vehicles 5, 6, 7

39

V. CONCLUSIONS AND RECOMMENDATIONS

In this final chapter, we review the work we have done on the problem of routing and scheduling TACAIR RECCE vehicles, and list our conclusions from this work. We also make recommendations on possible implementation of the results of our work, and on possible future areas of research on this problem.

A. CONCLUSIONS

As stated earlier in this paper, the Marine Corps will experience a growth in both types and numbers of TACAIR RECCE vehicles in the near future. Along with this increased potential capability is the need to be able to effectively and efficiently employ these new systems. One way to do this is to increase the quality, measured as a function of the target priority assignments, of the routes and schedules for these increasing number of vehicle assets. We sought to develop a model to help the Marine Corps do this, and with the heuristic based system we feel that we have accomplished this goal. Our system is capable of accepting as input the various target and vehicle data, along with the commander's and staff guidance concerning target priority, and then developing a good solution to the routing and scheduling problem for these vehicles.

Implementing such a system is not particularly difficult from a computational standpoint. The algorithms that we have developed are easily transferable to any computer programming language. In addition, the data flow requirements are minimal, as the only data needed by the scheduling and routing system consists of data already available and used by those involved in the present manually based

40

system. The collation of this information into an integrated target list/vehicle data/scheduling system is technically very simple.

However, in order to fully realize the potential of this automated routing and scheduling system, the Marine Corps will need to alter its present method of target allocation somewhat, by changing the level of command at which most of the scheduling for TACAIR RECCE takes place. Specifically, the Marine Corps will need to consider scheduling assets at the same level of command at which the assets are tasked, that is, at the higher headquarters directing the tactical vehicle subunits. By doing so, the Marine Corps will be able to derive maximum benefit from our routing and scheduling system, since a pre-allocation of targets to subunits will not need to be done prior to the time when individual target-to-vehicle assignments are made. The implementation of a system such as this should result not only in higher quality schedules, especially in the face of a greatly expanded target list, but also in reduced manpower resources being devoted to the task of routing and scheduling TACAIR RECCE vehicles.

B. FURTHER RESEARCH

Rarely, if ever, is work on a heuristic approach ever considered "completed" since the algorithm developer is seldom satisfied with the status of a solution. Our problem proves to be no exception. A number of possible ways to improve the heuristic seem both plausible and potentially rewarding, and we present several of these here.

First, we feel that our use of the *Orienteering Problem* (OP) model for this problem is correct, and should be extended by incorporating more of the most recent ideas concerning its solution into our heuristic. Golden et al [Ref. 14 and Ref. 15] propose several ideas for solving the basic OP model more effectively with

41

a heuristic. These ideas, covered in Chapter 2 of our paper, might provide more effective solutions. Especially notable are the ideas of center-of-gravity and learning. The challenge is to adapt these ideas to a multiple vehicle, time window constrained environment like our problem.

Another way to help the heuristic find better solutions might be to involve the user on an interactive basis. Reductions in run times for the heuristic due to the "jump start" given by the man-in-the-loop might be possible by allowing the user to manually establish the first few targets on each route. This would allow for more candidate schedules being formulated in the same amount of time. Also, this arrangement might allow for "what-iffing" a potential change in target or vehicle data, a situation not at all easily handled by the present manual system.

Finally, we feel that a hybrid system might hold the most potential of all. It would consist of this or a similar heuristic generating the input routes that are used as columns for a set packing model like that given by Desrosiers et al [Ref. 8]. Then, the resulting selected routes from the set packing model could be used as the starting point for the generation of a new batch of routes (columns) with the heuristic model. By continuing this alternation between models, using output from one as the input for the other, one might be able to derive the maximum benefits possible from these two different approaches. Therefore, we recommend this as a possible line of future research to anyone interested in the *Multi-Player Orienteering Problem with Time Windows* model.

APPENDIX A   HEURISTIC DATA SET DEVELOPMENT

In this appendix we give the method for developing the test data set used for the heuristic solution for both the target list and the vehicle input.

Combining the information obtained from discussions with Marine Corps representatives [Ref. 2] with other published guidance from the Department of Defense [Ref. 1], we designed a spreadsheet program that took as input the various parameters and returned to us a random number based data set. The various parameters and the assumed distributions are shown in TABLE 2. Note that distance is measured in unspecified units from the depot, located at X,Y coordinates (0,0). This is because the travel times of the vehicles from target to target are a function of the different vehicles' speeds, with conversion handled within the heuristic. Time values are given in minutes.

Input data for the vehicles was also developed, but this was much simpler since at most only two items needed to be stated for each vehicle, and these are user input. First, it is required that each utilized vehicle have as input its maximum routing time (corresponding to airborne time adjusted for reserve fuel safety factors). In addition, those vehicles which will have a "forced initialization" (see Chapter III, Section C.3) will require the target name and the initial launch point (or drop off point, in the case of air-launched vehicles) of the "forced initialization" target to be given. If the initial launch point is the depot, then it will be given as such.

The example input data set used for this thesis presentation consists of 150 targets and seven vehicles.

TABLE 2. HEURISTIC DATA SET DEVELOPMENT PARAMETERS

Parameter(s)	Program Variable Name	Distribution or Value
Target Position	X,Y	$X \approx$ Uniform (-25,25) $Y \approx$ Uniform (0,50)
Earliest Target Start Service Time	E_i	0
Latest Target Start Service Time	L_i	1200
Target Time Window Length	$L_i - E_i$	Uniform (5,240)
Target Service Time	S_i	Min(Uniform (1,50), $(L_i - E_i)$)
Target Priority	P_i	Order Statistic from Uniform (0,1) assigned to each target i, $i = 1$ to $NUMTGT$

1. TARGET DATA

The data listed is in the following format, read across rows, with one row per target:

> Target Name, X Coordinate, Y Coordinate, Service Time, Earliest Arrival Time, Latest Departure Time, Priority.

Note that for this example, all input values for X and Y Coordinates were reduced by 20 percent via an adjustment factor within the FORTRAN code. This adjustment factor makes for easy adjustment of the input data for code testing and research purposes, and is a user adjustable input parameter.

```
1.,11.,-13.,33.,655.,889.,80.
2.,45.,-18.,15.,1126.,1156.,95.
3.,42.,2.,28.,465.,548.,126.
4.,46.,-14.,31.,1060.,1202.,84.
5.,25.,2.,1.,1096.,1103.,41.
6.,43.,-16.,26.,1241.,1472.,43.
7.,34.,23.,48.,999.,1188.,77.
8.,5.,-22.,30.,1215.,1400.,51.
```

9.,14.,3.,13.,536.,630.,88.
10.,27.,24.,41.,345.,544.,50.
11.,2.,-20.,18.,352.,486.,87.
12.,6.,-25.,7.,1155.,1168.,102.
13.,8.,-12.,40.,512.,736.,48.
14.,38.,17.,44.,915.,1096.,69.
15.,42.,-24.,22.,655.,698.,140.
16.,33.,-2.,10.,50.,70.,9.
17.,22.,8.,10.,683.,703.,78.
18.,31.,0.,29.,1215.,1289.,60.
19.,18.,-19.,30.,307.,403.,59.
20.,13.,23.,5.,470.,677.,97.
21.,15.,-13.,35.,1379.,1582.,44.
22.,48.,9.,14.,1096.,1237.,111.
23.,46.,-3.,33.,1050.,1223.,123.
24.,16.,2.,23.,521.,577.,49.
25.,37.,-14.,4.,833.,841.,75.
26.,11.,0.,8.,1314.,1337.,109.
27.,14.,-24.,4.,893.,1062.,64.
28.,47.,16.,50.,522.,745.,124.
29.,10.,21.,7.,881.,1105.,70.
30.,30.,24.,44.,699.,874.,54.
31.,34.,-20.,9.,549.,601.,42.
32.,25.,-1.,3.,523.,529.,119.
33.,26.,13.,16.,1379.,1529.,7.
34.,15.,-1.,12.,106.,261.,45.
35.,43.,22.,45.,1329.,1439.,121.
36.,17.,-17.,46.,681.,887.,86.
37.,12.,-15.,34.,1043.,1138.,1.
38.,9.,21.,48.,264.,389.,5.
39.,25.,-6.,9.,1205.,1223.,106.
40.,2.,7.,25.,101.,151.,108.
41.,39.,12.,22.,1268.,1313.,35.
42.,17.,1.,7.,1228.,1242.,104.
43.,39.,4.,24.,336.,502.,14.
44.,47.,-11.,20.,332.,391.,134.
45.,47.,-10.,3.,898.,1028.,147.
46.,48.,8.,29.,797.,881.,37.
47.,36.,8.,9.,185.,292.,142.
48.,45.,10.,48.,819.,963.,52.
49.,25.,6.,47.,512.,688.,130.
50.,10.,24.,14.,390.,543.,100.
51.,10.,-1.,24.,24.,219.,19.
52.,47.,-3.,20.,173.,392.,55.
53.,25.,-11.,5.,1020.,1062.,93.
54.,22.,13.,30.,1153.,1278.,74.
55.,34.,-3.,24.,346.,411.,113.
56.,6.,22.,21.,1335.,1377.,114.
57.,1.,16.,46.,922.,1128.,143.
58.,50.,-1.,15.,672.,730.,144.
59.,21.,11.,8.,536.,552.,17.
60.,25.,18.,15.,1278.,1432.,23.
61.,26.,12.,16.,958.,1039.,13.
62.,4.,3.,49.,1096.,1201.,133.
63.,16.,4.,36.,1137.,1320.,25.

```
64.,14.,-9.,34.,849.,1056.,103.
65.,29.,1.,31.,177.,353.,115.
66.,48.,-9.,12.,125.,254.,24.
67.,25.,-13.,10.,547.,568.,58.
68.,21.,-25.,28.,263.,318.,91.
69.,19.,-19.,7.,676.,871.,139.
70.,2.,16.,40.,600.,681.,120.
71.,13.,-4.,38.,1319.,1402.,72.
72.,45.,-16.,25.,726.,830.,27.
73.,44.,-6.,22.,1262.,1487.,117.
74.,13.,6.,30.,618.,679.,12.
75.,11.,-9.,33.,847.,929.,129.
76.,15.,-25.,18.,971.,1161.,148.
77.,27.,-16.,3.,119.,125.,116.
78.,49.,-19.,34.,116.,248.,66.
79.,5.,10.,30.,266.,339.,29.
80.,27.,19.,20.,321.,515.,31.
81.,49.,-18.,30.,1389.,1449.,85.
82.,43.,-6.,3.,833.,839.,30.
83.,22.,-12.,33.,1360.,1509.,82.
84.,6.,22.,31.,244.,421.,99.
85.,28.,5.,24.,1026.,1226.,57.
86.,34.,-24.,20.,489.,613.,15.
87.,5.,21.,8.,813.,829.,136.
88.,41.,-10.,3.,648.,813.,2.
89.,10.,6.,12.,1377.,1405.,39.
90.,39.,6.,25.,773.,855.,150.
91.,16.,3.,21.,1346.,1387.,34.
92.,38.,-17.,27.,1323.,1382.,61.
93.,2.,-3.,15.,257.,487.,94.
94.,13.,-19.,31.,696.,758.,62.
95.,16.,6.,5.,517.,605.,79.
96.,34.,7.,36.,494.,566.,68.
97.,31.,6.,30.,207.,325.,89.
98.,10.,18.,13.,1340.,1474.,125.
99.,7.,-10.,37.,1186.,1349.,46.
100.,29.,-5.,27.,584.,760.,105.
101.,21.,3.,4.,547.,661.,22.
102.,5.,16.,24.,903.,951.,53.
103.,23.,-2.,28.,1412.,1485.,10.
104.,48.,-3.,6.,1252.,1463.,107.
105.,13.,25.,49.,427.,606.,56.
106.,2.,9.,32.,1012.,1203.,26.
107.,9.,-24.,32.,1342.,1522.,141.
108.,39.,-15.,24.,1067.,1294.,138.
109.,19.,-14.,38.,876.,1114.,145.
110.,17.,12.,2.,7.,107.,101.
111.,26.,-13.,12.,815.,959.,21.
112.,35.,-2.,40.,651.,873.,65.
113.,43.,-9.,41.,80.,314.,33.
114.,46.,-7.,5.,1207.,1432.,81.
115.,41.,10.,17.,679.,761.,36.
116.,38.,22.,15.,1211.,1293.,38.
117.,4.,19.,15.,1037.,1068.,47.
118.,46.,8.,40.,764.,844.,83.
```

```
119.,50.,-1.,18.,451.,584.,98.
120.,30.,-18.,44.,58.,232.,73.
121.,33.,4.,42.,1102.,1221.,131.
122.,36.,-9.,9.,703.,722.,3.
123.,5.,20.,31.,131.,370.,32.
124.,24.,20.,22.,881.,924.,110.
125.,8.,7.,41.,1203.,1363.,118.
126.,22.,11.,39.,1244.,1322.,4.
127.,17.,16.,8.,623.,677.,137.
128.,34.,0.,3.,1145.,1206.,20.
129.,4.,-7.,24.,772.,981.,90.
130.,5.,-15.,22.,1262.,1366.,96.
131.,46.,-13.,40.,380.,504.,92.
132.,21.,19.,6.,541.,666.,40.
133.,1.,0.,4.,513.,713.,6.
134.,18.,-17.,6.,603.,615.,132.
135.,20.,21.,11.,1091.,1147.,76.
136.,45.,19.,31.,1125.,1186.,135.
137.,15.,16.,30.,1096.,1290.,8.
138.,27.,1.,21.,1374.,1521.,128.
139.,47.,24.,41.,1037.,1201.,16.
140.,30.,15.,11.,195.,217.,149.
141.,21.,-14.,21.,876.,971.,127.
142.,14.,10.,3.,314.,487.,112.
143.,17.,-21.,13.,1030.,1181.,71.
144.,23.,9.,30.,990.,1162.,63.
145.,23.,17.,3.,192.,200.,28.
146.,25.,22.,43.,377.,526.,18.
147.,3.,-5.,22.,112.,344.,67.
148.,8.,-2.,21.,789.,904.,11.
149.,6.,18.,31.,277.,338.,122.
150.,27.,-19.,18.,146.,182.,146.
```

2. VEHICLE INPUT DATA

The vehicle input data used for the example problem is given below, according

to the following format :

Vehicle Name, Vehicle Airborne Time, "Forced Initialization" (equals 0 if none), X and Y Coordinates of the Drop Off or Launch Point for "Forced Initialization" Vehicles.

1	250.	3	30.	2.
2	250.	8	0.	20.
3	200.	0	0.	0.
4	200.	0	0.	4.
5	200.	0	0.	0.
6	200.	0	0.	0.
7	200.	0	0.	0.

APPENDIX B   HEURISTIC SOLUTION OUTPUT

This appendix contains an abbreviated sample of the intermediate output and the final solution output for our test problem.

1. INTERMEDIATE OUTPUT

The intermediate output shown is for the sample problem's first candidate schedule. It shows how each route is developed, by listing the targets in order as they are added to each route. Then, the entry into the routine **IMPROV** is shown, with the movement of one target (in this case) between vehicles.

```
BEGINNING RUN NUMBER    1 OUT OF    50
 150 TOTAL TARGETS
   7 TOTAL VEHICLES
X FACTOR =     .80  Y FACTOR =     .80  VEH FACTOR =   1.00

INITIALIZE VEHICLE     1 WITH TARGET    3
  CHOICE IS             32 FOR VEH            1
  CHOICE IS             95 FOR VEH            1
  CHOICE IS              9 FOR VEH            1
  CHOICE IS             20 FOR VEH            1
  CHOICE IS            132 FOR VEH            1
  CHOICE IS            127 FOR VEH            1
  CHOICE IS            101 FOR VEH            1
  CHOICE IS            100 FOR VEH            1
  CHOICE IS            133 FOR VEH            1

INITIALIZE VEHICLE     2 WITH TARGET    8
  CHOICE IS             26 FOR VEH            2
  CHOICE IS            130 FOR VEH            2
  CHOICE IS            107 FOR VEH            2
  CHOICE IS            138 FOR VEH            2
  CHOICE IS             73 FOR VEH            2

 Init. No. = 3
INITIALIZE VEHICLE     3 WITH TARGET   76
  CHOICE IS             27 FOR VEH            3
  CHOICE IS             53 FOR VEH            3
  CHOICE IS            143 FOR VEH            3
  CHOICE IS            141 FOR VEH            3
  CHOICE IS             75 FOR VEH            3

 Init. No. = 3
```

48

```
INITIALIZE VEHICLE      4 WITH TARGET    45
 CHOICE IS             69 FOR VEH               4
 CHOICE IS             87 FOR VEH               4
 CHOICE IS            109 FOR VEH               4
 CHOICE IS             29 FOR VEH               4

 Init. No. = 4
INITIALIZE VEHICLE      5 WITH TARGET    58
 CHOICE IS             15 FOR VEH               5
 CHOICE IS            134 FOR VEH               5
 CHOICE IS            115 FOR VEH               5
 CHOICE IS             88 FOR VEH               5

 Init. No. = 1
INITIALIZE VEHICLE      6 WITH TARGET    90
 CHOICE IS             25 FOR VEH               6
 CHOICE IS            129 FOR VEH               6
 CHOICE IS             64 FOR VEH               6
 CHOICE IS            111 FOR VEH               6

 Init. No. = 3
INITIALIZE VEHICLE      7 WITH TARGET    57
 CHOICE IS             62 FOR VEH               7
 CHOICE IS             39 FOR VEH               7
 CHOICE IS             42 FOR VEH               7
 CHOICE IS            128 FOR VEH               7
 IN SUBROUTINE IMPROV NOW
 CHOICE IS            133 FOR VEH               5
 EXIT CANSCED

NEW BEST OVERALL RUN HAS 4129.0 TOTAL POINTS

BEGINNING RUN NUMBER   2 OUT OF    50
```

.

.

.

2. FINAL SOLUTION OUTPUT

Shown here is the final solution output from the FORTRAN program of our heuristic solution. It takes the form of a simplified flight schedule, listing first the pertinent vehicle data, following with that vehicle's target assignments and the target's input and computed data. This output could very easily be used to set up the

unit's flight schedule for the time period involved. The variables shown are defined
as follows (except for those defined in Chapter 3 of the thesis text) :

VT	Vehicle scheduled airborne time (time aloft)
VSCORE	Sum of target priority points for vehicle
VLAUN	Vehicle launch time

```
BEST OVERALL RUN HAS 4659.0 TOTAL POINTS

RESULTS AND SCHEDULES

VEHICLE NUMBER    1  VTMAX =  250.0  VT =  248.1  VSCORE =   819.0
        VLAUN =  450.6       SLIDEF =    .0  SLIDEB =     .0
  TARGET NUMBER  151  Arrive at =  450.6  Depart at =  450.6  P =      .0
    EARLIEST =  441.4  LATEST = 9999.0  MAXPF =     .0  MAXPB =   4.9
    TGTVEH =    1
    .X =   24.0  Y =    1.6  W =      .0  S =      .0
  TARGET NUMBER    3  Arrive at =  484.2  Depart at =  512.2  P =  126.0
    EARLIEST =  465.0  LATEST =  548.0  MAXPF =     .0  MAXPB =   4.9
    TGTVEH =    1
     X =   33.6  Y =    1.6  W =      .0  S =   28.0
  TARGET NUMBER   32  Arrive at =  526.0  Depart at =  529.0  P =  119.0
    EARLIEST =  523.0  LATEST =  529.0  MAXPF =     .0  MAXPB =   3.0
    TGTVEH =    1
     X =   20.0  Y =    -.8  W =      .0  S =    3.0
  TARGET NUMBER   95  Arrive at =  538.1  Depart at =  543.1  P =   79.0
    EARLIEST =  517.0  LATEST =  605.0  MAXPF =    1.2  MAXPB =   3.0
    TGTVEH =    1
     X =   12.8  Y =    4.8  W =      .0  S =    5.0
  TARGET NUMBER    9  Arrive at =  546.0  Depart at =  559.0  P =   88.0
    EARLIEST =  536.0  LATEST =  630.0  MAXPF =    1.2  MAXPB =   3.0
    TGTVEH =    1
     X =   11.2  Y =    2.4  W =      .0  S =   13.0
  TARGET NUMBER  132  Arrive at =  573.0  Depart at =  579.0  P =   40.0
    EARLIEST =  541.0  LATEST =  666.0  MAXPF =    1.2  MAXPB =   3.0
    TGTVEH =    1
     X =   16.8  Y =   15.2  W =      .0  S =    6.0
  TARGET NUMBER   20  Arrive at =  586.1  Depart at =  591.1  P =   97.0
    EARLIEST =  470.0  LATEST =  677.0  MAXPF =    1.2  MAXPB =   3.0
    TGTVEH =    1
     X =   10.4  Y =   18.4  W =      .0  S =    5.0
  TARGET NUMBER  101  Arrive at =  608.4  Depart at =  612.4  P =   22.0
    EARLIEST =  547.0  LATEST =  661.0  MAXPF =    1.2  MAXPB =   3.0
    TGTVEH =    1
     X =   16.8  Y =    2.4  W =      .0  S =    4.0
  TARGET NUMBER  100  Arrive at =  621.4  Depart at =  648.4  P =  105.0
    EARLIEST =  584.0  LATEST =  760.0  MAXPF =    1.2  MAXPB =   3.0
    TGTVEH =    1
```

```
        X =    23.2  Y =    -4.0  W =      .0  S =    27.0
    TARGET NUMBER  127  Arrive at =  667.8  Depart at =  675.8  P =   137.0
      EARLIEST =  623.0  LATEST =  677.0  MAXPF =     1.2  MAXPB =     3.0
     TGTVEH =    1
        X =    13.6  Y =    12.8  W =      .0  S =     8.0
    TARGET NUMBER  133  Arrive at =  693.9  Depart at =  697.9  P =     6.0
      EARLIEST =  513.0  LATEST =  713.0  MAXPF =     1.9  MAXPB =     3.0
     TGTVEH =    1
        X =      .8  Y =      .0  W =      .0  S =     4.0
    TARGET NUMBER  152  Arrive at =  698.7  Depart at =  698.7  P =       .0
      EARLIEST =      .0  LATEST =  700.6  MAXPF =     1.9  MAXPB =     3.0
     TGTVEH =    1
        X =      .0  Y =      .0  W =      .0  S =      .0


VEHICLE NUMBER     2  VTMAX =   250.0  VT =   233.7  VSCORE =   642.0
        VLAUN = 1305.2       SLIDEF =      .0  SLIDEB =      .0
    TARGET NUMBER  153  Arrive at = 1305.2  Depart at = 1305.2  P =      .0
      EARLIEST = 1288.9  LATEST = 1352.0  MAXPF =     3.7  MAXPB =    16.3
 TGTVEH =    2
        X =    0 .0  Y =    16.0  W =      .0  S =      .0
    TARGET NUMBER   26  Arrive at = 1314.0  Depart at = 1322.0  P =   109.0
      EARLIEST = 1314.0  LATEST = 1337.0  MAXPF =     3.7  MAXPB =      .0
 TGTVEH =    2
        X =     8.8  Y =      .0  W =      .0  S =     8.0
    TARGET NUMBER  130  Arrive at = 1334.9  Depart at = 1356.9  P =    96.0
      EARLIEST = 1262.0  LATEST = 1366.0  MAXPF =     3.7  MAXPB =      .0
 TGTVEH =    2
        X =     4.0  Y =   -12.0  W =      .0  S =    22.0
    TARGET NUMBER    8  Arrive at = 1362.5  Depart at = 1392.5  P =    51.0
      EARLIEST = 1215.0  LATEST = 1400.0  MAXPF =     3.7  MAXPB =      .0
 TGTVEH =    2
        X =     4.0  Y =   -17.6  W =      .0  S =    30.0
    TARGET NUMBER  107  Arrive at = 1396.1  Depart at = 1428.1  P =   141.0
      EARLIEST = 1342.0  LATEST = 1522.0  MAXPF =     3.7  MAXPB =      .0
 TGTVEH =    2
        X =     7.2  Y =   -19.2  W =      .0  S =    32.0
    TARGET NUMBER   73  Arrive at = 1459.6  Depart at = 1481.6  P =   117.0
      EARLIEST = 1262.0  LATEST = 1487.0  MAXPF =     3.7  MAXPB =      .0
 TGTVEH =    2
        X =    35.2  Y =    -4.8  W =      .0  S =    22.0
    TARGET NUMBER  138  Arrive at = 1496.3  Depart at = 1517.3  P =   128.0
      EARLIEST = 1374.0  LATEST = 1521.0  MAXPF =     3.7  MAXPB =      .0
 TGTVEH =    2
        X =    21.6  Y =      .8  W =      .0  S =    21.0
    TARGET NUMBER  154  Arrive at = 1538.9  Depart at = 1538.9  P =      .0
      EARLIEST = 1263.0  LATEST = 1558.9  MAXPF =    20.0  MAXPB =      .0
 TGTVEH =    2
        X =      .0  Y =      .0  W =      .0  S =      .0


VEHICLE NUMBER     3  VTMAX =   200.0  VT =   169.2  VSCORE =   637.0
        VLAUN =  171.6       SLIDEF =      .0  SLIDEB =      .0
    TARGET NUMBER  155  Arrive at =  171.6  Depart at =  171.6  P =      .0
      EARLIEST =  757.0  LATEST =  798.4  MAXPF =      .0  MAXPB =    33.3
```

51

```
TGTVEH =    3
   X =      .0  Y =      .0  W =      .0  S =      .0
  TARGET NUMBER  145  Arrive at =  194.5  Depart at =  197.5  P =    28.0
    EARLIEST =  192.0  LATEST =  200.0  MAXPF =      .0  MAXPB =     2.5
TGTVEH =    3
   X =    18.4  Y =    13.6  W =      .0  S =     3.0
  TARGET NUMBER  140  Arrive at =  203.3  Depart at =  214.3  P =   149.0
    EARLIEST =  195.0  LATEST =  217.0  MAXPF =      .0  MAXPB =     2.5
TGTVEH =    3
   X =    24.0  Y =    12.0  W =      .0  S =    11.0
  TARGET NUMBER   47  Arrive at =  221.7  Depart at =  230.7  P =   142.0
    EARLIEST =  185.0  LATEST =  292.0  MAXPF =      .0  MAXPB =     2.5
TGTVEH =    3
   X =    28.8  Y =     6.4  W =      .0  S =     9.0
  TARGET NUMBER   34  Arrive at =  249.0  Depart at =  261.0  P =    45.0
    EARLIEST =  106.0  LATEST =  261.0  MAXPF =      .0  MAXPB =     2.5
TGTVEH =    3
   X =    12.0  Y =     -.8  W =      .0  S =    12.0
  TARGET NUMBER  147  Arrive at =  271.1  Depart at =  293.1  P =    67.0
    EARLIEST =  112.0  LATEST =  344.0  MAXPF =    30.8  MAXPB =     2.5
TGTVEH =    3
   X =     2.4  Y =    -4.0  W =      .0  S =    22.0
  TARGET NUMBER   93  Arrive at =  294.9  Depart at =  309.9  P =    94.0
    EARLIEST =  257.0  LATEST =  487.0  MAXPF =    30.8  MAXPB =     2.5
TGTVEH =    3
   X =     1.6  Y =    -2.4  W =      .0  S =    15.0
  TARGET NUMBER  142  Arrive at =  324.1  Depart at =  327.1  P =   112.0
    EARLIEST =  314.0  LATEST =  487.0  MAXPF =    30.8  MAXPB =     2.5
TGTVEH =    3
   X =    11.2  Y =     8.0  W =      .0  S =     3.0
  TARGET NUMBER  156  Arrive at =  340.8  Depart at =  340.8  P =      .0
    EARLIEST =  829.6  LATEST =  975.9  MAXPF =    30.8  MAXPB =     2.5
TGTVEH =    3
   X =      .0  Y =      .0  W =      .0  S =      .0


VEHICLE NUMBER    4  VTMAX =  200.0  VT =  199.3  VSCORE =   604.0
     VLAUN =  764.7      SLIDEF =      .0  SLIDEB =      .0
  TARGET NUMBER  157  Arrive at =  764.7  Depart at =  764.7  P =      .0
    EARLIEST =  138.3  LATEST =  179.2  MAXPF =      .0  MAXPB =     4.7
TGTVEH =    4
   X =      .0  Y =     4.0  W =      .0  S =      .0
  TARGET NUMBER   90  Arrive at =  795.9  Depart at =  820.9  P =   150.0
    EARLIEST =  773.0  LATEST =  855.0  MAXPF =      .0  MAXPB =     4.7
TGTVEH =    4
   X =    31.2  Y =     4.8  W =      .0  S =    25.0
  TARGET NUMBER   25  Arrive at =  837.0  Depart at =  841.0  P =    75.0
    EARLIEST =  833.0  LATEST =  841.0  MAXPF =      .0  MAXPB =     4.0
TGTVEH =    4
   X =    29.6  Y =   -11.2  W =      .0  S =     4.0
  TARGET NUMBER   69  Arrive at =  855.9  Depart at =  862.9  P =   139.0
    EARLIEST =  676.0  LATEST =  871.0  MAXPF =      .7  MAXPB =     4.0
TGTVEH =    4
   X =    15.2  Y =   -15.2  W =      .0  S =     7.0
  TARGET NUMBER  111  Arrive at =  870.3  Depart at =  882.3  P =    21.0
```

```
        EARLIEST =  815.0  LATEST =  959.0  MAXPF =        .7  MAXPB =     4.0
TGTVEH =    4
     X =   20.8  Y =  -10.4  W =      .0  S =   12.0
   TARGET NUMBER   75  Arrive at =  894.7  Depart at =  927.7  P =  129.0
     EARLIEST =  847.0  LATEST =  929.0  MAXPF =        .7  MAXPB =     4.0
TGTVEH =    4
     X =    8.8  Y =   -7.2  W =      .0  S =   33.0
   TARGET NUMBER  129  Arrive at =  933.6  Depart at =  957.6  P =   90.0
     EARLIEST =  772.0  LATEST =  981.0  MAXPF =        .7  MAXPB =     4.0
TGTVEH =    4
     X =    3.2  Y =   -5.6  W =      .0  S =   24.0
   TARGET NUMBER  158  Arrive at =  964.0  Depart at =  964.0  P =      .0
     EARLIEST =  232.8  LATEST =  373.8  MAXPF =        .7  MAXPB =     4.0
TGTVEH =    4
     X =      .0  Y =      .0  W =      .0  S =      .0


VEHICLE NUMBER    5  VTMAX =  200.0  VT =  196.0  VSCORE =  706.0
       VLAUN =  986.6       SLIDEF =      .0  SLIDEB =      .0
   TARGET NUMBER  159  Arrive at =  986.6  Depart at =  986.6  P =      .0
     EARLIEST =  982.6  LATEST = 1057.1  MAXPF =        .0  MAXPB =     4.0
TGTVEH =    5
     X =      .0  Y =      .0  W =      .0  S =      .0
   TARGET NUMBER   45  Arrive at = 1025.0  Depart at = 1028.0  P =  147.0
     EARLIEST =  898.0  LATEST = 1028.0  MAXPF =        .0  MAXPB =     4.0
TGTVEH =    5
     X =   37.6  Y =   -8.0  W =      .0  S =    3.0
   TARGET NUMBER   53  Arrive at = 1045.6  Depart at = 1050.6  P =   93.0
     EARLIEST = 1020.0  LATEST = 1062.0  MAXPF =   11.4  MAXPB =     4.0
TGTVEH =    5
     X =   20.0  Y =   -8.8  W =      .0  S =    5.0
   TARGET NUMBER  109  Arrive at = 1056.0  Depart at = 1094.0  P =  145.0
     EARLIEST =  876.0  LATEST = 1114.0  MAXPF =   17.4  MAXPB =     4.0
TGTVEH =    5
     X =   15.2  Y =  -11.2  W =      .0  S =   38.0
   TARGET NUMBER  143  Arrive at = 1099.8  Depart at = 1112.8  P =   71.0
     EARLIEST = 1030.0  LATEST = 1181.0  MAXPF =   17.4  MAXPB =     4.0
TGTVEH =    5
     X =   13.6  Y =  -16.8  W =      .0  S =   13.0
   TARGET NUMBER   76  Arrive at = 1116.4  Depart at = 1134.4  P =  148.0
     EARLIEST =  971.0  LATEST = 1161.0  MAXPF =   17.4  MAXPB =     4.0
TGTVEH =    5
     X =   12.0  Y =  -20.0  W =      .0  S =   18.0
   TARGET NUMBER   12  Arrive at = 1141.6  Depart at = 1162.0  P =  102.0
     EARLIEST = 1155.0  LATEST = 1168.0  MAXPF =   17.4  MAXPB =      .0
TGTVEH =    5
     X =    4.8  Y =  -20.0  W =   13.4  S =    7.0
   TARGET NUMBER  160  Arrive at = 1182.6  Depart at = 1182.6  P =      .0
     EARLIEST =  932.9  LATEST = 1186.6  MAXPF =    4.0  MAXPB =      .0
TGTVEH =    5
     X =      .0  Y =      .0  W =      .0  S =      .0


VEHICLE NUMBER    6  VTMAX =  200.0  VT =  191.1  VSCORE =  567.0
       VLAUN =   79.6       SLIDEF =      .0  SLIDEB =      .0
```

```
TARGET NUMBER  161  Arrive at =   79.6  Depart at =   79.6  P =    .0
   EARLIEST =  578.5  LATEST =  675.0  MAXPF =     .0  MAXPB =    8.9
TGTVEH =    6
   X =     .0  Y =     .0  W =     .0  S =     .0
TARGET NUMBER  110  Arrive at =   96.2  Depart at =   98.2  P =  101.0
   EARLIEST =    7.0  LATEST =  107.0  MAXPF =     .0  MAXPB =    8.9
TGTVEH =    6
   X =   13.6  Y =    9.6  W =     .0  S =    2.0
TARGET NUMBER   77  Arrive at =  122.0  Depart at =  125.0  P =  116.0
   EARLIEST =  119.0  LATEST =  125.0  MAXPF =     .0  MAXPB =    3.0
TGTVEH =    6
   X =   21.6  Y =  -12.8  W =     .0  S =    3.0
TARGET NUMBER  150  Arrive at =  127.4  Depart at =  164.0  P =  146.0
   EARLIEST =  146.0  LATEST =  182.0  MAXPF =   27.5  MAXPB =    .0
TGTVEH =    6
   X =   21.6  Y =  -15.2  W =   18.6  S =   18.0
TARGET NUMBER   65  Arrive at =  180.1  Depart at =  211.1  P =  115.0
   EARLIEST =  177.0  LATEST =  353.0  MAXPF =    8.9  MAXPB =    .0
TGTVEH =    6
   X =   23.2  Y =     .8  W =     .0  S =   31.0
TARGET NUMBER   97  Arrive at =  215.4  Depart at =  245.4  P =   89.0
   EARLIEST =  207.0  LATEST =  325.0  MAXPF =    8.9  MAXPB =    .0
TGTVEH =    6
   X =   24.8  Y =    4.8  W =     .0  S =   30.0
TARGET NUMBER  162  Arrive at =  270.6  Depart at =  270.6  P =    .0
   EARLIEST =  727.0  LATEST =  785.0  MAXPF =    8.9  MAXPB =    .0
TGTVEH =    6
   X =     .0  Y =     .0  W =     .0  S =     .0


VEHICLE NUMBER    7  VTMAX =  200.0  VT =  195.3  VSCORE =   684.0
       VLAUN = 1193.4      SLIDEF =     .0  SLIDEB =     .0
TARGET NUMBER  163  Arrive at = 1193.4  Depart at = 1193.4  P =    .0
   EARLIEST = 1048.6  LATEST = 1115.9  MAXPF =     .0  MAXPB =    8.2
TGTVEH =    7
   X =     .0  Y =     .0  W =     .0  S =     .0
TARGET NUMBER   39  Arrive at = 1214.0  Depart at = 1223.0  P =  106.0
   EARLIEST = 1205.0  LATEST = 1223.0  MAXPF =     .0  MAXPB =    8.2
TGTVEH =    7
   X =   20.0  Y =   -4.8  W =     .0  S =    9.0
TARGET NUMBER   42  Arrive at = 1231.5  Depart at = 1238.5  P =  104.0
   EARLIEST = 1228.0  LATEST = 1242.0  MAXPF =    3.5  MAXPB =    3.5
TGTVEH =    7
   X =   13.6  Y =     .8  W =     .0  S =    7.0
TARGET NUMBER  108  Arrive at = 1260.3  Depart at = 1284.3  P =  138.0
   EARLIEST = 1067.0  LATEST = 1294.0  MAXPF =    4.7  MAXPB =    3.5
TGTVEH =    7
   X =   31.2  Y =  -12.0  W =     .0  S =   24.0
TARGET NUMBER  114  Arrive at = 1292.8  Depart at = 1297.8  P =   81.0
   EARLIEST = 1207.0  LATEST = 1432.0  MAXPF =    4.7  MAXPB =    3.5
TGTVEH =    7
   X =   36.8  Y =   -5.6  W =     .0  S =    5.0
TARGET NUMBER  104  Arrive at = 1301.3  Depart at = 1307.3  P =  107.0
   EARLIEST = 1252.0  LATEST = 1463.0  MAXPF =    4.7  MAXPB =    3.5
TGTVEH =    7
```

```
      X =    38.4  Y =    -2.4  W =      .0  S =     6.0
   TARGET NUMBER   60  Arrive at = 1332.3  Depart at = 1347.3  P =    23.0
      EARLIEST = 1278.0  LATEST = 1432.0  MAXPF =     4.7  MAXPB =    3.5
TGTVEH =    7
      X =    20.0  Y =    14.4  W =      .0  S =    15.0
   TARGET NUMBER   98  Arrive at = 1359.3  Depart at = 1372.3  P =   125.0
      EARLIEST = 1340.0  LATEST = 1474.0  MAXPF =     4.7  MAXPB =    3.5
TGTVEH =    7
      X =     8.0  Y =    14.4  W =      .0  S =    13.0
   TARGET NUMBER  164  Arrive at = 1388.7  Depart at = 1388.7  P =     .0
      EARLIEST = 1195.1  LATEST = 1257.0  MAXPF =     4.7  MAXPB =    3.5
TGTVEH =    7
      X =      .0  Y =      .0  W =      .0  S =      .0
 Time taken     +1491.1833496 secs
```

LIST OF REFERENCES

1. U.S. Department of Defense, *Unmanned Aerial Vehicle Master Plan*, Government Printing Office, pp. III-1 - III-5, Washington, DC, 1989.

2. Interviews between D. Kestner, Captain, USMC, Marine Aviation Weapons and Tactics Squadron One, MCAS Yuma, AZ, and the author, 10 - 14 December 1989 and 23-24 June 1990.

3. Solomon, M., and Desrosiers, J., "Time Window Constrained Routing and Scheduling Problems," *Transportation Science*, v. 22, no. 1, pp. 1-13, February 1988.

4. Christofides, N., and others, eds. , *Combinatorial Optimization*, John Wiley & Sons, 1979.

5. Bodin, L., and others, *Routing and Scheduling of Vehicles and Crews - The State of the Art*, Pegamon Press, 1983.

6. Golden, B. L., and Assad, A. A., eds., *Vehicle Routing: Methods and Studies,* North-Holland, 1988.

7. Kolen, A. W. J., Rinnooy Kan, A. H. G., and Trienkens, H. W. J. M., "Vehicle Routing with Time Windows," *Operations Research*, v. 35, no. 2, pp. 266-273, March-April 1987.

8. Desrosiers, J., Soumis, F., and Desrochers, M., "Routing with Time Windows by Column Generation," *Networks*, v. 14, pp. 545-565, 1984.

9. Solomon, M. M., "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Research*, v. 35 no. 2, March-April 1987.

10. Lawler, E. L., and others, eds., *The Traveling Salesman Problem*, John Wiley and Sons, 1985.

11. Baker, E. K., "An Exact Algorithm for the Time-Constrained Traveling Salesman Problem," *Operations Research*, v. 31, no. 5, pp. 938-945, September-October 1983.

12. Volgenant, T., and Jonker, R., "On Some Generalizations of the Travelling-Salesman Problem," *Journal of the Operations Research Society*, v. 38, no. 11, pp 1073-1079, 1987.

13. Tsiligirides, T., "Heuristic Methods Applied to Orienteering," *Journal of the Operational Research Society*, v. 35 no. 9, pp. 797-809, 1984.

14. Golden, B. L., Levy, L., and Vohra, R., "The Orienteering Problem," *Naval Research Logistics*, v. 34, pp. 307-318, 1987.

15. Golden, B. L., Wang, Q., and Liu L., "A Multifaceted Heuristic for the Orienteering Problem," *Naval Research Logistics*, v. 35, pp. 359-366, 1988.

16. Ramesh, R., Yoon, Y. S., and Karwan, M. H., *An Optimal Algorithm for the Orienteering Tour Problem Using Reformulation and Lagrangean Relaxation*, Working paper Number 710, School of Management, State University of New York at Buffalo, June 1989.

17. Schrage, L., *Linear, Integer and Quadratic Programming with LINDO*, Third Edition, The Scientific Press, 1986.

18. Chun, B. J., and Lee, S. H., *Algorithms and Heuristics for Time-Window-Constrained Traveling Salesman Problems*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1985.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Superintendent
   Attn: Library, Code 52   2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Professor R. E. Rosenthal   2
   Department of Operations Research, Code OR/Rl
   Naval Postgraduate School
   Monterey, California 93943

4. Captain P. S. Bloch, USN   1
   Department of Operations Research, Code OR/Bl
   Naval Postgraduate School
   Monterey, California 93943

5. Commanding Officer   1
   Marine Aviation Weapons and Tactics Squadron One
   MCAS Yuma, Arizona 85369

6. Navy Tactical Support Activity   1
   Washington Navy Yard
   Washington, D.C. 20380

7. Captain H. D. Moser, Jr.   2
   2011 W. 13 Lane
   Yuma, Arizona 85364